DESIGN AND FABRICATION OF A CODEC FOR DIGITAL TROPO

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

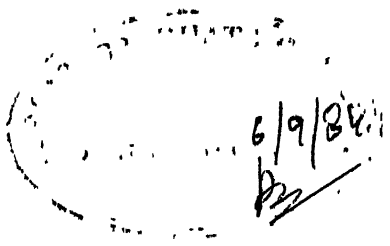
by

VASIMALLA JEEVAN GAGARIN

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
SEPTEMBER, 1984

21 St[illegible] F.E. 3584.- M. GAC.- D.P.S.



CERTIFICATE

Cerified that the work reported in the thesis titled 'Design and Fabrication of a Codec for Digital Tropo' has been carried out under our supervision by Mr. Vasimalla Jeevan Gagarin and has not been submitted elsewhere for a degree.

K.R. Sarma
Professor
Dept. of Electrical Engineering
Indian Institute of Technology
Kanpur

P.R.K. Rao
Professor
Dept. of Electrical Engineering
Indian Institute of Technology
Kanpur

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude to Dr. P.V.V. Rao and Dr. K.S. Sarma for their invaluable guidance and constant encouragement throughout the course of this work.

I thank Dr. J. Das, Dr. V.P. Sinha, Dr. P.V. Chatterjee and Dr. M.U. Siddiqui for teaching me the material which formed the background for this attempt.

I also thankfully appreciate the help extended to me by Mr. Joseph John, Research Engineer on different occasions of need. I thank Mr. P.V. Rao for his help at various stages of my thesis.

I greatly acknowledge Dr. P. Rayaratnam and family for their help and encouragement throughout my stay in this place.

I thank all my friends for making my stay in this campus memorable.

I thank Mrs. Rukmini for her excellent typing.

Jeevan Gaganin

ABSTRACT

The design and fabrication of an encoder and decoder for digital communication over a troposcatter channel for a fade rate of 10 Hz has been considered.

The performance of BCH codes (31,6), (31,11), (31,16), (31,21), (31,26), (15,5), (15,7) and (15,11), single error correcting and adjacent double error correcting codes (7,3) and (8,4), and extened Hamming codes (8,4), (8,3) and (8,2) has been evaluated without and with interleaving for the cases of no-explicit diversity and dual diversity by simulating the slowly fading channel on DEC-1090 system.

The extended Hamming codes (8,4) and (8,2) have been implemented. The input data rate for (8,4) code is 1 Mbps. This (8,4) code is capable of correcting all single errors and 6 out of 7 adjacent double errors. (8,2) code is used for low data rates. The input data rate for (8,2) code is 500 kbps. It corrects all single errors, adjacent double and triple errors, and many random errors. A simple scheme is used to achieve word synchronization in the decoder.

CONTENTS

Page

1. Introduction

1.1 The Problem and the Approach - - - - - - - - - - - - - - 1
1.2 Organization of the Thesis - - - - - - - - - - - - - - - 3

2. Coding Schemes for Tropo - - - - - - - - - - - - - - - - - 5

2.1 Specifications of a Typical Digital Mobile Tropo System - 5

2.1.1 System Parameters - - - - - - - - - - - - - - - - 6
2.1.2 System Requirements - - - - - - - - - - - - - - - 6

2.2 Characteristics of Coding Schemes under Consideration - 9

2.2.1 Random Error Correcting or Detecting Codes - - 9
2.2.2 Burst Error Correction - - - - - - - - - - - - - 11

2.3 Time-Interleaving - - - - - - - - - - - - - - - - - - - 12
2.4 Multiple Error Correcting Codes - - - - - - - - - - - 16

2.4.1 BCH Codes - - - - - - - - - - - - - - - - - - - 16
2.4.2 SEC-DAEC Codes - - - - - - - - - - - - - - - - - 17
2.4.3 Extended Hamming Codes - - - - - - - - - - - - - 19

2.5 Description of BCH Codes - - - - - - - - - - - - - - - 20

2.5.1 Construction of BCH Codes - - - - - - - - - - - 21
2.5.2 Decoding of BCH Codes - - - - - - - - - - - - - 24

2.6 Description of SEC-DAEC Codes - - - - - - - - - - - - 29
2.7 Description of Extended Hamming Codes - - - - - - - - 31

3. Performance Evaluation of the Coding Schemes - - - - - - 35

3.1 Evaluation of the Performance of the Coding Schemes 35

3.1.1 Encoding - 36
3.1.2 Transmitting the Binary Sequence over the Fading Channel - - - - - - - - - - - - - - - - - 36
3.1.3 Demodulation - - - - - - - - - - - - - - - - - - 38
3.1.4 Decoding - 39
3.1.5 Results - 41

4. Hardware Implementation of a Codec - - - - - - - - - - - 63

4.1 Description of the Encoder - - - - - - - - - - - - - - - 64

4.1.1 Implementation of the Encoder - - - - - - - - - 66
4.1.2 Clock Generator - - - - - - - - - - - - - - - - 68

4.2 Description of Decoder - - - - - - - - - - - - - - - - 69

4.2.1 Implementation of the Decoder - - - - - - - - - 69
4.2.2 Word Synchronizer - - - - - - - - - - - - - - - 72

5. Conclusions - 75

6. References - 79

Chapter 1

INTRODUCTION

1.1 The Problem and the Approach

As a part of an overall effort directed towards establishing all digital communication networks, there is a growing interest in digital communications over satellite systems and in the conversion of land based line of sight (LOS) microwave systems into digital systems. Troposcatter systems form an integral part of the present communication systems and they provide an important alternative to the land based LOS because of the larger troposcatter link span. These systems operate over large stretches of water or inaccessable terrain by covering more distance in one link, thereby reducing the manning and site requirements to fewer terminal locations. This is particularly true when the problem of protecting the installations is acute.

Single-hop troposcatter systems are capable of establishing over the horizon radio communications between mobile terminals with link lengths upto 300 kms. But the severe handicap of this mobile digital communication system is the limitation on the transmitter power capability. For low bit-error rate performance, using the conventional schemes of signalling to combat the effects of fading and multipath characteristics of tropo-channels, these systems require large transmitter power levels and explicit diversity reception[1].

Signal fading is due to time varying multipath phenomenon exhibited by the channel and dispersion is due to multipath. Fading results in the loss of information and multipath leads to intersymbol interference.

In order to combat the effects of noise and fading, one can strive for invulnerability to their effects through appropriately designed redundancy, i.e., through coding when data is coded in order to check the above effects, parity check digits are added to every block of message digits such that errors occuring in the codeword can be detected and corrected by the decoder. The reduction of the SNR per digit due to the addition of parity check digits to the stream of message digits is more than offset by the reduction in the bit error rate.

In troposcatter communication systems, typical fade duration is 100 milliseconds. The error phenomenon associated with fading is one of bursts with lengths up to a few thousands of bits. Because errors occuring in the channel are of burst nature, the coding schemes that are being employed to detect and correct these errors must be of burst-error correction type. Unless these burst-error correcting codes have lengths far greater than the fading span, this burst phenomenon can not be solved. Techniques to effectively increase the code-span must then be considered along with their practicability. The effects of burst errors can be obviated by a process of interleaving, in which adjacent bits are spaced in time to avoid being caught in the same fade. Therefore time-interleaving which randomises these

burst errors is the best alternative. Using multiple random error correcting codes, errors occuring in the data stream due to fading can be limited.

In the digital communication system under consideration, binary data at 1 M bps bit rate enters the transmitter which consists of an encoder, an interleaver and a modulator. These coded blocks are interleaved before they enter the modulator. When data is interleaved at the transmitter, a corresponding deinterleaver is used at the receiver end to bring together the shuffled symbols of the same codeword before decoding.

This thesis considers an integrated coding-interleaving approach for digital communications over a Rayleigh fading channel. The main aim is to illustrate the performance gains that can be achieved by such an approach. BCH, SEC-DAEC codes (modified Hamming codes) and extended Hamming codes are employed to illustrate the benifits of this integrated approach.

1.2 Organization of the Thesis

In Chapter 2, theoretical and practical considerations of BCH codes, modified Hamming codes and extended Hamming codes that are being considered are discussed.

In Chapter 3, the performance evaluation of the coding schemes proposed in Chapter 2 is carried out in terms of the bit error probability versus received SNR per information bit for the case of slowly-fading channel by means of computer simulation.

Simulation is carried out for both cases of no-explicit diversity and dual diversity, in each case with interleaving and without interleaving. Results are illustrated through tables and graphs.

Chapter 4 gives the implementation considerations of codecs. Codes that are implemented are extended Hamming (8,4) and (8,2) codes. A simple technique is used for word synchronization.

Chapter 5 concludes the thesis with suggestions and recommendations for further work.

Chapter 2

CODING SCHEMES FOR TROPO

In this chapter, specifications of a typical digital mobile tropo-system are given in the first section. For these specifications the system using coding-interleavig approach achieves the required average bit error rate. This coding-interleaving approach requires the use of efficient error correcting coding schemes. Characteristics of these error-correcting codes, both random and burst error correcting, are presented in the second section. Description of time-interleaving technique is given in the third section. Features of error-correcting codes under consideration are given in the fourth section and their description is given in the subsequent sections.

2.1 Specifications of a Typical Digital Mobile Tropo System

In this section, specifications of a digital mobile tropo system, link parameters and path losses are given.

For different values of link lengths, the data rates to be achieved are shown in Table 2.1.

2.1.1 System Parameters

(a) Fixed parameters

Transmitter power = p_t = 200 W (53 dbm)

Antenna gain = G = 84 db

Feeder loss (both sides) = l_f = 2 db

Receiver noise figure = 3 db

Carrier frequency = f_c = 4.7 GHz

(b) Details of power amplifier

Klystron with 400 W and 5 MHz BW

TWT with 200 W and 40 MHz BW

(c) Options available

(i) Explicit dual diversity (if necessary)

(ii) Mode of transmission : BPSK or QPSK

2.1.2 System Requirements

Average probability of bit error = $P_b(\epsilon) = 10^{-3}$

Time availability = 99 %

Confidence level = 85 %

For these specifications, path losses for different link lengths are known. They are shown in Table 2.2.

Table 2.1

Link Data Specifications

Distances to be covered (kms)	Input information data rate (Kbps)
120	1024
160	512
240	256
320	128

Table 2.2

Path Loss

Link length (kms)	Path loss (dbs)
120	232.3
160	236.7
240	242.5
320	249.5

The available (E_b/N_o) per diversity has been calculated for all combinations of distances and data rates of interest[2]. These values are shown in Table 2.3.

Table 2.3

Available $(\overline{E_b/N_o})$/diversity

Link length (kms)	Path loss (db)	P_r (db)	R=1/T (kbps)	Noise power (db)	Available E_b/N_o/div (db)
120	232.3	-97.3	128	-119.90	24.2
			256	-116.90	21.2
			512	-113.90	18.2
			1024	-110.90	15.2
			2048	-107.90	12.2
160	236.7	-101.7	128	-119.90	19.8
			256	-116.90	16.8
			512	-113.90	13.8
			1024	-110.90	10.8
			2048	-107.90	7.8
240	242.5	-107.5	128	-119.90	14.0
			256	-116.90	11.0
			512	-113.90	8.0
			1024	-110.90	5.0
			2048	-107.90	2.0
320	249.5	-114.5	128	-119.90	7.0
			256	-116.90	4.0
			512	-113.90	1.0
			1024	-110.90	-2.0
			2048	-107.90	-5.0

For these specifications, the average value of $(\overline{E_b/N_o})$ required to achieve the specified average probability of error rate over fading tropo scatter channel using methods like time gating or DFE, is larger than the available value of $(\overline{E_b/N_o})$ at the receiver. In order to achieve the required performance, signal power may be increased. In a mobile communication system, the transmitter power is severely limited. Therefore, increasing the bit energy is not the appropriate solution.

It has been shown that by coding-interleaving approach, employing dual diversity, the required value of $(\overline{E_b/N_o})$ can be decreased for the given error rate[2]. But coding results in the increase of bandwidth. In this application, the available bandwidth is 5 MHz and the required value is 2 MHz. Therefore bandwidth is not a constraint in this case.

2.2 Characteristics of Coding Schemes under Consideration

Error control codes for binary case may be broadly classified into two major groups, block codes and non-block codes. Block codes are further classified into cyclic and non-cyclic codes. Convolutional codes are an example of non-block codes. In this thesis, convolutional codes are not considered because hardware realization of convolutional codes is more complex compared to block codes.

This section gives an account on the error correction or detection capability of block coding schemes for random and burst

errors. A block code of length n with k message digits, which corrects t errors is expressed as an (n,k) code.

Rate or efficiency of an (n,k) code = k/n

2.2.1 Random Error Correcting or Detecting Codes

For an (n,k) code, a total of 2^n codewords are available in an n-dimensional binary space to be assigned to 2^k message words. A message word $\underline{m}_i$ may be transmitted in a coded form as $\underline{c}_i$. Because of channel errors, the received word will not be $\underline{c}_i$, and is indicated as $\underline{c}'_i$. If the channnel noise causes errors in t or less digits, then $\underline{c}'_i$ will lie somewhere in the Hamming sphere of radius t centred at $\underline{c}_i$. If t errors are to be corrected, the coding scheme must have property that all of the Hamming spheres of radius t centred at the code-words are non-overlapping. It means, one may not use words of n-dimension that lie within a Hamming distance of t from any code word. If a received word lies within a Hamming sphere of radius t centred at $\underline{c}_i$, then $\underline{c}_i$ is decoded as the transmitted code word. This scheme is capable of correcting upto t-errors and d_{min}, the minimum distance between any two t-error correcting code words is 2t + 1.

The number of sequences of n digits that differ from a given sequence by j digits is given by ${}^nc_j$, j = 1,2, ... t. Hence the number of ways in which up to t-errors can occur is given by $\sum_{j=1}^{t} \binom{n}{j}$. Hence for each codeword $\sum_{j=1}^{t} \binom{n}{j}$ number of words must not be used since there are 2^k code words, one must leave $2^t \sum_{j=1}^{t} \binom{n}{j}$

words unused. Therefore the total number of words available within the space must be at least $2^k + 2^k \sum_{j=1}^{t} \binom{n}{j} = 2^k \sum_{j=0}^{t} \binom{n}{j}$. The total number of words available is 2^n. Thus,

$$2^n \geq 2^k \sum_{j=0}^{t} \binom{n}{j}$$

or

$$2^{n-k} \geq \sum_{j=0}^{t} \binom{n}{j}$$

$$2^m \geq \sum_{j=0}^{t} \binom{n}{j}$$

where, m = n-k is the number of parity checks required. This is known as the Hamming bound. Hamming bound is necessary but not a sufficient condition. If some m satisfies the Hamming bound, it does not mean that a t-error correcting code of n digits can be constructed.

Another way of correcting errors is to use a code only to detect (not to correct) up to t-errors. When the receiver detects an error, it requests for retransmission. Because error-detection requires fewer check digits, these codes operate at a higher rate.

To detect t-errors, code words must be separated by a Hamming distance of not more than t + 1. Suppose a transmitted code word c_i has x number of errors, and $x \leq t$, then the received code word $\underline{c'_i}$ is at a distance of x from $\underline{c_i}$. Since $x \leq t$, $\underline{c'_i}$ can never be another code word, since all code words are separated by at least a distance of t + 1. Thus, the reception of $\underline{c'_i}$ immediately indicates that an error has occured and

requests for retransmission. But this method requires a feed back path in order to transmit the repeat request.

2.2.2 Burst Error Correcting Codes

Burst errors are those that wipe out some or all of a sequence of digits. In general, random-error-correcting codes are not efficient for correcting burst-errors. Hence special burst error-correcting codes are found.

A burst of length b is defined as a sequence of digits in which the first and the b^{th} digits are in error, and the (b-2) digits in between may or may not be in error.

It has been shown that for detecting all burst errors of length b or less with a linear block code of length a, b parity check bits are necessary and sufficient[8].

If a received sequence of length b or less is in error, parity will be violated and errors will be detected but not corrected, and the receiver can request for retransmission of the b digits lost. Therefore a linear code with b parity checks detects not only all bursts of length b or less, but also a high percentage of longer bursts.

For correcting burst errors of length b, a linear block code must have at least 2b parity checks[8].

In order to correct all bursts of length b or less and simultaneously detect all bursts of length $p > b$, the code must have at least $b + p$ parity checks[8].

2.3 Time-Interleaving

In tropo-scatter communications, the channel under consideration fades occasionally. The typical fade duration is 100 msecs. These fades results in bursts of errors of length of the order of a few thousands of bits. If such burst errors are to be controlled, using burst error correcting codes, the number of parity checks required is double to that of the burst length. The code length of such burst-error-correcting codes will be of the order of a few thousands of bits. They are not practicable. An alternative to check this burst error phenomenon is to separate every two adjacent bits by a fade duration. But this results in the reduction of channel capacity. For high data rates this is not appropriate.

Using a technique known as time-interleaving every two adjacent bits in a codeword can be spaced in time by (1/B) seconds, which is equal to the typical fade duration and bits of other codewords are interleaved into this guard space. Thus data is transmitted maintaining guard space betwen every two adjacent bits so that even if a burst occcrs, no two bits of a codeword are lost. Thus, burst errors are randomized. Using random error correcting codes, these errors can be corrected, thereby achieving the required value of error rate performance.

The structure of interleaver and deinterleaver are given in Figure 2.2. The overall system model is given in Figure 2.3.

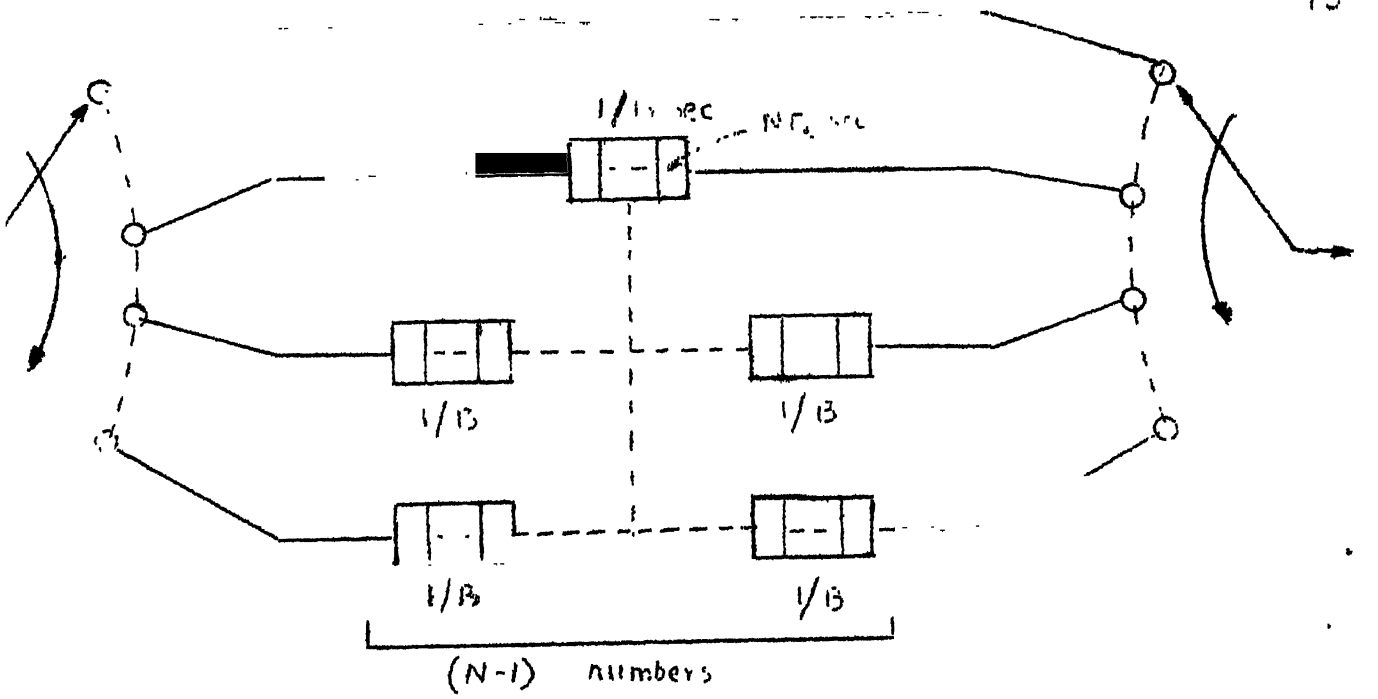


Fig 2.1a. THE INTERLEAVER

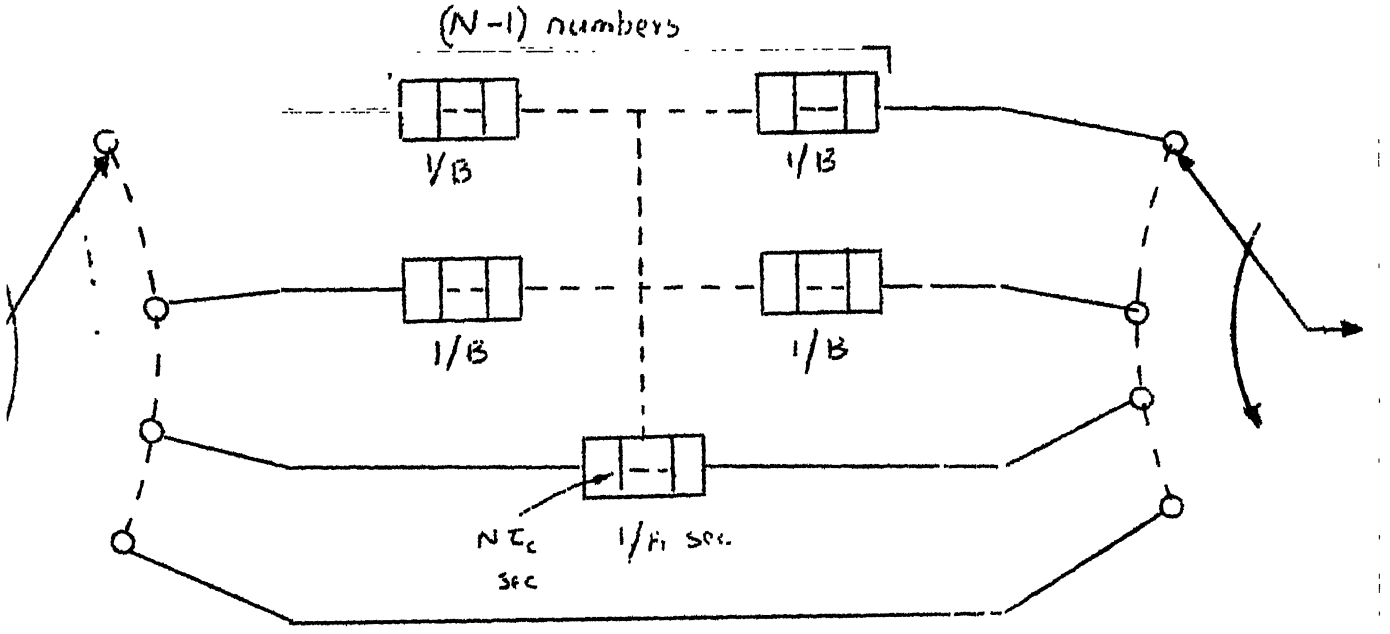


Fig 2.1b THE DEINTERLEAVER

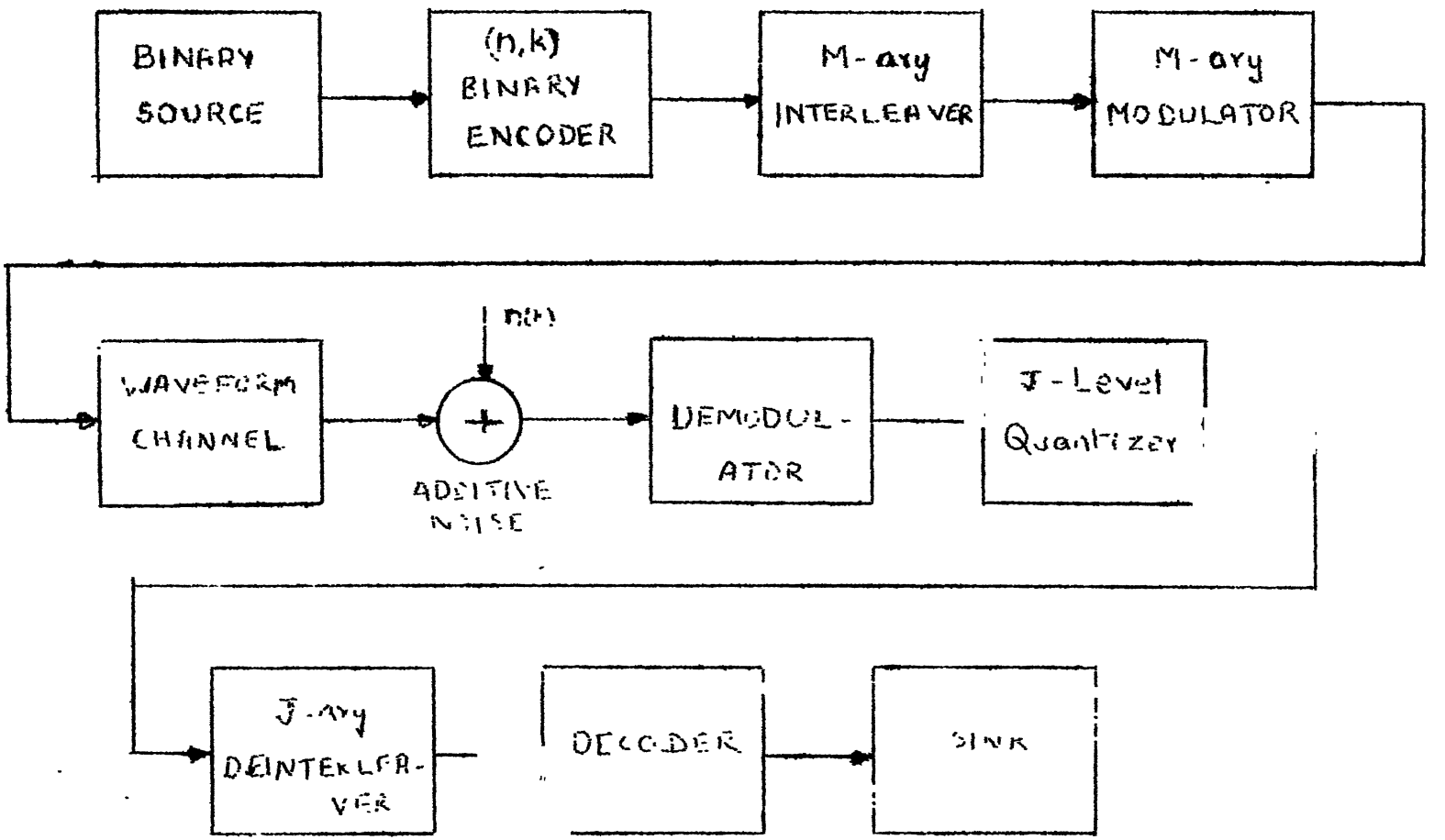


FIG ... MODEL OF A DIGITAL COMMUNICATION SYSTEM

In the following section, multiple error correcting codes are considered.

2.4 Multiple Error Correcting Codes

In this section, both cyclic and non-cyclic codes are considered. Cyclic codes that are being considered are BCH, and single-error-correcting and double adjacent error correcting (SEC-DAEC) codes. For non-cyclic codes, extended Hamming codes are considered.

2.4.1 BCH Codes

It has been found[2] that codes of length 31, which can correct about 5 to 7 errors are quite attractive in tropo-communication applications. BCH (31,6) and (31,11) codes are capable of correcting 7 and 5 errors respectively. In case of (31,6) code, even if 7 symbols of any code word fall in a deep fade, due to its error-correcting capability, the decoder would be able to extract the transmitted information bits. Also, this reduces the basic time-interleaving duration by a factor of 7, which in turn reduces the memory storage requirements to 1/7 of the actual value.

The interleaver/deinterleaver combination of (31,6) code requires about 1 M bits of memory, and it is very complex to organize such a large amount of memory. Because of this practical difficulty, other coding schemes of shorter length are

considered. But in order to evaluate the performance of BCH codes over slowly fading channel, BCH codes of length 31 and 15 are considered.

2.4.2 SEC-DAEC Codes

It has been observed that whenever a deep fade occurs, there is a possibility of losing two adjacent bits, but within a code word, except for single errors, random errors generally do not occur. In this situation, SEC-DAEC codes suggested by Abramson[9] are very much useful.

A complete-double error correcting code increases the reliability of the transmission at the cost of greatly increased number of parity check digits necessary to transmit each word. Compared to these codes, SEC-DAEC codes uses fewer check digits.

Because of inherennt regularities in the structure of these codes, they can be implemented with remarkable simplicity using shift registers. For the purpose of comparison, the maximum number of information digits possible (k*) for the class of SEC-DAEC codes, annd the maximum number of information digits possible (k**) for codes which correct all double errors, for a given number of parity checks, m = 1,2,... 10 are shown in Table 2.4.

Table 2.4

The Number of Information Digits for SEC-DAEC and SEC-DEC Codes

m	k*	k**
1	-	-
2	-	-
3	0	-
4	3	1
5	10	2
6	25	4
7	56	8
8	119	14
9	256	22
10	501	34

The number of parity checks necessary for any given number of information digits can be obtained from the following relation. An (n,k) code with k message digits and m parity checks can correct 2^m error patterns. In this case, there are (k+ m) possible single errors, (k + m) possible adjacent double errors, an error in first and last digits of the code is also considered to be a double adjacent error and finally the possibility of no errors. Therefore,

$$2^m >= 2(m + k) + 1$$

Since the left side of the inequality is even and the right side is odd, a one may be added to the right without destroying the inequality.

$$2^m >= 2(m + k) + 2 = 2(m + k + 1)$$

$$2^{m-1} >= m + k + 1$$

or $$k <= 2^{m-1} - m - 1$$

For a given m, k* is the maximum value that can be obtained. An (n,k) code with k < = k* and m = n-k, is known as a SEC-DAEC code.

From this class of codes, (7,3) and (9,4) are considered. The (7,3) code requires a memory of 350 kbits for interleaver/deinterleaver structure, and the (9,4) code requires 450 kbits. These schemes can be realized without much complexity.

2.4.3 Extended Hamming Codes

This class of codes are non-cyclic and they are known for their error-correcting capability. In this thesis, codes of length 8 are considered.

The extended (8,4) Hamming code is capable of correcting 6 out of 7 adjacent double errors besides all single errors. This may be used for high data rates. The interleaver/deinterleaver structure requires 350 kbits of memory.

The extended (8,3) Hamming code is capable of correcting all single errors, and all adjacent double and triple errors. One configuration of (8,2) code is capable of correcting all single errors and bursts of all lengths, i.e., from 2 to 8. The same combination can be used to correct all single errors, all adjacent double and triple errors and many random double and triple errors. Here burst means all bits within a burst of

length b are in error. These codes are easy to implement. Depending on the data rate, one can switch over from scheme to another (8,4) may be used for high data rates and (8,2) for low data rates and the same interleaver/deinterleaver structure may be used for all these schemes, keeping the transmitter output bit rate constant.

In the following section BCH codes are described.

2.5 Description of BCH Codes

BCH codes are cyclic codes. For any positive integers m and t, ($t < 2^{m-1}$), there exists a Bose-Chaudhuri-Hocquenghem (BCH) code with the following parameters.

block length $n = 2^{m-1}$

number of parity checks $n-k < mt$

minimum distance $d >= 2t + 1$

This code is capable of correcting any combination of t or fewer errors in a block of $n = 2^{m-1}$ digits. The generator polynomial of this code is derived as follows.

Let α be a primitive element of the Galois field $GF(2^m)$. Let $m_i(x)$ be the minimal polynomial of α^i. Then the generator polynomial of the t-error-correcting BCH code is[8]

$$g(x) = LCM(m_1(x), m_2(x), \ldots, m_{2t}(x)) \qquad 2.5.1$$

Clearly, $\alpha, \alpha^2, \ldots, \alpha^{2t}$ are roots of g(x), i.e., $g(\alpha^i) = 0$ for $i = 1,2, \ldots 2t$. If i is an even integer, it

can be expressed as a product of the following form:

$$i = i' \, 2^{-p}$$

where i' is an odd integer and $p >= 1$. Then α^i, $\alpha^{i'}$ have the same minimal polynomial, i.e.,

$$m_i(x) = m_{i'}(x) \qquad 2.5.2$$

Therefore, every even power of α in the sequence of α, α^2, ..., α^{2t} has the same minimal polynomial as some previous odd power of α in the sequence. As a result, the generator polynomial of the t-error-correcting BCH is given by Equn. 2.5.1 reduces to

$$g(x) = \text{LCM}\ (m_1(x), m_3(x), \ldots, m_{2t-1}(x)) \qquad 2.5.3$$

Since the degree of each minimal polynomial is m or less, the degree of g(x) is at most mt. In other words, the number of parity-check digits, n-k, is at most equal to mt. These BCH codes generated by g(x) are called primitive BCH codes.

2.5.1 Construction of BCH Codes

Let α be a primitive element of the Galois field $GF(2^5)$ and $m_1(x)$, $m_3(x)$, $m_5(x)$, $m_7(x)$, $m_{11}(x)$ and $m_{15}(x)$ be the minimal polynomials of α, α^3, α^5, α^7, α^{11} and α^{15} respectively. To find any minimal polynomial the following sequence has to be formulated

$$\alpha = \alpha^2 = \alpha^4 = \alpha^8 = \alpha^{16}$$
$$\alpha^3 = \alpha^6 = \alpha^{12} = \alpha^{24} = \alpha^{48-31} = \alpha^{17}$$

$\alpha^5 = \alpha^{10} = \alpha^{20} = \alpha^{40-31} = \alpha^9 = \alpha^{18}$

$\alpha^7 = \alpha^{14} = \alpha^{28} = \alpha^{56-31} = \alpha^{25} = \alpha^{50-31} = \alpha^{19}$

$\alpha^{11} = \alpha^{22} = \alpha^{44-31} = \alpha^{13} = \alpha^{26} = \alpha^{52-31} = \alpha^{21}$

$\alpha^{15} = \alpha^{30} = \alpha^{60-31} = \alpha^{29} = \alpha^{58-31} = \alpha^{27} = \alpha^{54-31} = \alpha^{23}$

thus, $m_1(x)$ has α, α^2, α^4, α^8 and α^{16} as all its roots, and

$$m_1(x) = (x + \alpha)(x + \alpha^2)(x + \alpha^4)(x + \alpha^8)(x + \alpha^{16})$$
$$= 1 + x^2 + x^5$$

$$m_3(x) = (x + \alpha^3)(x + x^6)(x + \alpha^{12})(x + \alpha^{24})(x + \alpha^{17})$$
$$= 1 + x^2 + x^3 + x^4 + x^5$$

$$m_5(x) = (x + \alpha^5)(x + \alpha^{10})(x + \alpha^{20})(x + \alpha^9)(x + \alpha^{18})$$
$$= 1 + x + x^2 + x^4 + x^5$$

$$m_7(x) = (x + \alpha^7)(x + \alpha^{14})(x + \alpha^{28})(x + \alpha^{25})(x + \alpha^{19})$$
$$= 1 + x + x^2 + x^3 + x^5$$

$$m_{11}(x) = (x + \alpha^{11})(x + \alpha^{22})(x + \alpha^{13})(x + \alpha^{26})(x + \alpha^{21})$$
$$= 1 + x + x^3 + x^4 + x^5$$

$$m_{15}(x) = (x + \alpha^{15})(x + \alpha^{30})(x + \alpha^{29})(x + \alpha^{27})(x + \alpha^{23}) = 1 + x^3 + x^5$$

Similarly for n = 15, or $GF(2^4)$, the other set of minimal polynomials are $m_1(x)$, $m_3(x)$, $m_5(x)$. Expressions for m_1, m_3 and m_5 are

$$m_1(x) = 1 + x + x^4$$
$$m_3(x) = 1 + x + x^2 + x^3 + x^4$$
$$m_5(x) = 1 + x + x^2$$

According to the Equn. 2.5.3 t-errors-correcting BCH codes of length $n = 2^5 - 1 = 31$ are obtained as shown below.

$t = 1, \quad g(x) = m_1(x) = 1 + x^2 + x^5$

$t = 2, \quad g(x) = LCM(m_1(x), m_3(x))$

$= 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$

$t = 3, \quad g(x) = LCM(m_1(x), m_3(x), m_5(x))$

$= 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + x^9 + x^{10}$

$+ x^{11} + x^{15}$

$t = 4,5 \quad g(x) = LCM(m_1(x), m_3(x), m_5(x), m_7(x))$

$= 1 + x^2 + x^4 + x^6 + x^7 + x^9 + x^{10} + x^{13}$

$+ x^{17} + x^{18} + x^{20}$

$t = 6,7 \quad g(x) = LCM(m_1(x), m_3(x), m_5(x), m_7(x), m_{11}(x)$

$= 1 + x + x^2 + x^5 + x^9 + x^{11} + x^{13} + x^{14}$

$+ x^{15} + x^{16} + x^{18} + x^{19} + x^{21} + x^{24} +$

$+ x^{25}$

$t = 8,9,10, \ldots, 15$

$g(x) = LCM(m_1(x), m_3(x), m_5(x), m_7(x), m_{11}(x),$

$m_{15}(x))$

$1 + \sum_{i=1}^{30} x^i$

Similarly, for $n = 2^4 - 1 = 15$, BCH codes are

$t = 1, \quad g(x) = m_1(x) = 1 + x + x^4$

$t = 2, \quad g(x) = LCM(m_1(x), m_3(x)) = 1 + x^4 + x^6 + x^7 + x^8$

$t = 3, \quad g(x) = LCM(m_1(x), m_3(x), m_5(x))$

$= 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$

In all these cases, minimum distance $d >= 2t + 1$. For $t = 1$, the BCH code of length $2^m - 1$ is generated by $g(x) = m_1(x)$.

Since α is a primitive element of GF (2^m), $m_1(x)$ is a primitive polynomial of degree m. Therefore, the Hamming code of length $n = 2^m - 1$ is the single-error-correcting BCH code of length $n = 2^m - 1$ for any $m >= 3$. Therefore, the Hamming codes constitute a subclass of the primitive BCH codes.

2.5.2 Decoding of the BCH codes

BCH codes are decoded in three steps.

1. Syndrome calculation
2. Error-locator polynomial generation
3. Error-correction

2.5.2.1 Syndrome Calculation

Let $V(x) = v_o + v_1x + v_2x^2 + \ldots + v_{n-1}x^{n-1}$ be the transmitted code vector and

$$R(x) = r_0 + r_1x + r_2x^2 + \ldots + r_{n-1}x^{n-1}$$

be the received vector. Then the error pattern added by the noisy channel is

$$e(x) = R(x) + V(x) \qquad 2.5.4$$

Syndrome is defined as a vector $\underline{S}$ with 2t components as follows.

$$S_i = r(\alpha^i)$$

$$= r_o + r_1 \alpha^i + r_2(\alpha^i)^2 + \ldots + r_{n-1}(\alpha^i)^{n-1} \qquad 2.5.5$$

for $i = 1,2, \ldots , 2t$.

Combining Eqns. 2.5.4 and 2.5.5

$$S_i = v(\alpha^i) + e(\alpha^i)$$

is obtained. Since α , α^2, ... , α^{2t} are roots of the code

polynomial v(x), then

$$S_i = e(\alpha^i) \qquad 2.5.6$$

for i = 1, 2, ... , 2t. It is assumed that e(x) is an error pattern of p errors, given by

$$e(x) = x^{j_1} + x^{j_2} + \ldots + x^{j_p} \qquad 2.5.7$$

Using eqn. 2.5.6, we get

$$\begin{aligned} S_1 &= \alpha^{j_1} + \alpha^{j_2} + \ldots + \alpha^{j_p} \\ S_2 &= (\alpha^{j_1})^2 + (\alpha^{j_2}) + \ldots + (\alpha^{j_p})^2 \\ S_3 &= (\alpha^{j_1})^3 + (\alpha^{j_2})^3 + \ldots + (\alpha^{j_p})^3 \\ \ldots\ S_{2t} &= (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \ldots + (\alpha^{j_p})^{2t} \end{aligned} \qquad 2.5.8$$

Any error correction procedure is a method of solving this set of equations for α^{j_1}, α^{j_2}, ... , α^{j_p} . Once α^{j_1}, α^{j_2} , ... ,α^{j_p} have been found, then the powers $j_1, j_2, \ldots, j_p$ will give the error locations in e(X) as in eqn. (2.5.7). In general, these equations have a finite number of solutions. Each solution yields a different error pattern. If the number of errors in the actual error pattern e(X) is t or less (p < t), then the solution which yields an error pattern with the smallest number of errors is the right solution. The error pattern corresponding to this solution is the actual error pattern e(x) caused by the channel noise. For large t, solving the above equations for α^{j1} is difficult and ineffective.

An effective procedure to determine α^{j1} from the syndrame components S_i's is given below.

The error location numbers are written as

$$\beta_l = \alpha^{j_l} \text{ for } 1 \leq l \leq p \qquad 2.5.9$$

Now 2.5.8 may be rewritten as

$$\begin{aligned} S_1 &= \beta_1 + \beta_2 + \ldots + \beta_p \\ S_2 &= \beta_1^2 + \beta_2^2 + \ldots + \beta_p^2 \\ S_{2t} &= \beta_1^{2t} + \beta_2^{2t} + \ldots + \beta_p^{2t} \end{aligned} \qquad 2.5.10$$

These 2t syndrome components are symmmetric functions in $\beta_1, \beta_2, \ldots \beta_p$, which are known as power-sum symmetric functions.

2.5.2.2 Error Locator Polynomial Generation

The error location polynomial is defined as follows:

$$\begin{aligned} \sigma(x) &= (1 + \beta_1 x)(1 + \beta_2 x)(1 + \beta_3 x) \ldots (1 + \beta_p x) \\ &= \sigma_o + \sigma_1 x + \sigma_2 x^2 + \ldots + \sigma_p x^p \end{aligned} \qquad 2.5.11$$

where

$$\begin{aligned} \sigma_o &= 1 \\ \sigma_1 &= \beta_1 + \beta_2 + \beta_3 + \ldots \beta_p \\ \sigma_2 &= \beta_1 \beta_2 + \beta_2 \beta_3 + \ldots + \beta_{p-1} \beta_p \\ \sigma_p &= \beta_1 \beta_2 \ldots \beta_p \end{aligned} \qquad 2.5.12$$

The roots of $\sigma(x)$ are $\beta_1^{-1}, \beta_2^{-1}, \ldots, \beta_p^{-1}$, which are the inverse of error location numbers. It is clear from eqns. (2.5.10) and (2.5.12) that the coefficients of $\sigma(x)$ are related to the syndrome components S_i, for $i = 1,2, \ldots, 2t$. Therefore,

it is possible to find $\sigma(x)$ from S' s, then the error location numbers can be found and the error pattern e(x) can be determined. The coefficients $\sigma_1, \sigma_2, \ldots, \sigma_p$ are known as elementary symmetric functions of $\beta_1 \beta_2, \ldots, \beta_r$ To generate error location polynomial $\sigma(x)$, an iterative algorithm given by Perlekamp is chosen. To find $\sigma(x)$, the table to be filled up is

μ	$\sigma^{(\mu)}(x)$	d_μ	l_μ	$2\mu-l\mu$
-1/2	1	1	0	1
0	1	S_1	0	0
1				
2				
.				
.				
.				
t				

Assuming that all rows up to and including the μ^{th} row, are filled the $(\mu+1)$ the row is filled out according to the rules given below.

(1) If $d_\mu = 0$, then $\sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x)$

(2) If $d_\mu \neq 0$, find another one preceding the μ^{th}, say the q^{th}, such that the number of 2q - lq in the last column is as large as possible and $dq \neq 0$. Then

$$\sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x) + d_\mu^{-1} d_q \, x^{2(\mu-q)} \sigma^{(q)}(x)$$

In either case, $\lambda_{\mu+1}$ is exactly the degree of $\sigma^{(\mu+1)}(x)$, and

$$d_{\mu+1} = S_{2\mu+3} + \sigma_1^{(\mu+1)} S_{2\mu+2} + \sigma_2^{(\mu+1)} S_{2\mu+1} + \cdots + \sigma_{\lambda_{\mu+1}}^{(\mu+1)} S_{2\mu+3-\lambda_{\mu+1}} \qquad 2.5$$

The polynomial $\sigma^t(x)$ in the last line is the required $\sigma(x)$. If it has degree greater than t, there are more than t errors, and it is not possible to locate them, or it may give erroneous locations.

2.5.2.3 Error Correction

This step involves calculation of erroneous locations and error correction. The error location numbers are nothing but reciprocals of the roots of $\sigma(x)$. The roots of $\sigma(x)$ are found by substituting $1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$ $(n = 2^m - 1)$ into $\sigma(x)$. Since $\alpha^n = 1$, $\alpha^{-1} = \alpha^{n-1}$. Therefore, if α^1 is a root of $\sigma(x)$, α^{n-1} is an error-location number and the received digit v_{n-1} is an erroneous digit.

2.6 Description of SEC-DAEC Codes

This class of single-error correcting and double adjacent error correcting codes are derived from cyclic single-error-correcting Hamming codes. These SEC-DAEC codes are also cyclic. They are obtained by converting one message bit of SEC codes into parity bit. A SEC-DAEC code with number of message digits less than k* information digits is constructed by obtaining the code

for k* information digits and then making the first few information digits equal to zero. But the number of parity checks remain the same. Because of the increase in ratio of k/(n-k) the error-correcting capability of this code increases. Besides all single and adjacent double errors, they can correct a few other error patterns.

The parity check equations of (7,3) code are:

$$y_1 = x_1 + x_2$$
$$y_2 = x_2 + x_3$$
$$y_3 = x_3 + y_1$$
$$y_4 = x_1 + x_2 + x_3 + y_1 + y_2 + y_3$$

The parity check equations of (15, 10) code are:

$$y_1 = x_1 + x_3 + x_6 + x_7 + x_9 + x_{10}$$
$$y_2 = x_2 + x_4 + x_7 + x_8 + x_{10} + y_1$$
$$y_3 = x_3 + x_5 + x_8 + x_9 + y_1 + y_2$$
$$y_4 = x_4 + x_6 + x_9 + x_{10} + y_2 + y_3$$
$$y_5 = \sum_{i=1}^{10} x_i + \sum_{i=1}^{4} y_i$$

From this (15, 10) code, (9,4) SEC-DAEC obtained by setting x_1 - x_6 to zero. The parity check equation of (9,4) code are:

$$y_1 = x_1 + x_3 + x_4$$
$$y_2 = x_1 + x_2 + x_4 + y_1$$
$$y_3 = x_2 + x_3 + y_1 + y_2$$
$$y_4 = x_3 + x_4 + y_2 + y_3$$
$$y_5 = \sum_{i=1}^{4} x_i + \sum_{i=1}^{4} y_i$$

These codes are inherently regular in structure[10] and they can be implemented with remarkable simplicity using shift registers.

Message words of length k are encoded using feed back shift registers of length m. The connections of the feed back shift register depends on the generator polynomial.

Received code words are decoded in three steps:

1. Syndrome generation
2. Error-pattern detection
3. Bit-error correction

The decoder mainly consists of a feed-back shift register (fbsr) and a buffer register. When a code vector is received, it is fed simultaneously to fbsr and buffer. The feed back connections of fbsr are the same as the one which performs encoding operation. It initially contains all zeros. While a received word is being decoded, no fresh word is accepted. If the received word is the same as the one transmitted, the contents of the fbsr will be zero, after all bits of the received vector are fed. If the received word is erroneous, the contents will not be zero. This process is known as syndrome generation. Encoder and decoder are shown in Figure 2.4.

Decoder is provided with error-pattern-detector circuit. This continuously checks the fbsr contents as the shifting continues. For every shift, the contents of fbsr changes and the digits of the uncorrected message leave the buffer. Whenever an

erroneous digit reaches the right end of the buffer (assuming that they are fed from left), the detector circuitary recognises the configuration of the fbrs and emits a one in the next shift. This one is added to the erroneous digit which comes out of the buffer at the same time a one is fed back to fbrs. This is continued until the entire received word leaves the buffer. Thus error-detection and error-correction is carried using shift registers. Implementation of SEC-DAEC codes using fbsr is given by Meggit[10].

2.7 Description of Extended Hamming Codes

These extended Hamming codes are non-cyclic block codes. These codes are derived from single error correcting Hamming codes. These codes correct all single errors and a few other burst error patterns. These codes are obtained by adding one or two parity digits or by converting message digits into parity digits. For a specified code length and error correcting requirement, these codes are obtained by an extensive search using computer.

In this case, codes of different error correcting capability are obtained for a code length of 8. These codes are (8,4), (8,3) and (8,2).

The generator matrix of (8,4) code which corrects all single errors and 6 out of 7 adjacent double errors is found to be

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

The generator matrix of (8,3) extended Hamming code which corrects all single errors, adjacent double errors and adjacent triple errors and a few other random error patterns is

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

The generator matrices of (8,2) codes which corrects all single errors and bursts of all lengths, and a few other random error pattern are shown below.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

First combination canalso be used to correct all single errors, all adjacent double and triple errors and many other random errors, instead of bursts of length 4, 5, 6, 7, 8 which are generally not likely.

A code vector is a linear combination of the rows of G. For an (n,k) code, G is of dimension (k x n). If m is a message vector of dimension k, there are 2^k possible message vectors and

each message vector can be encoded into a unique vector of dimension n.

The SEC-DAEC and extended Hamming codes considered have very small values of k and n, and it is easy to encode a message vector by multiplying it with the generator matrix G.

If $\underline{u}$ the transmitted vector and $\underline{v}$ is the received vector corresponding to $\underline{u}$, then the error pattern added by the noisy channel is,

$$\underline{e} = \underline{u} + \underline{v}$$

To decode SEC-DAEC and extended Hamming codes, table-look-up technique may be used, as k values are very small. The standard-array required for decoding is formulated as follows.

Let c_1 be the identity element and c_2, c_3, ... , c_{2k} are the other code vectors. These code vectors are placed in a row with the identity element at the extreme left. Next, of all the remaining n-types, a chosen one say c_1 is placed under the identity element. This would be one of the most likely vector to be received if the identity vector is transmitted. Then the row is completed by placing under each code vector c_i, the vector $c_i + e_1$. Similarly a second vector e_2 is placed in the first column and the row is completed. The process is cotinued until all possible n-types appear some where in the array. The rows are cosets and the vectors in the first column are coset leaders.

If the standard array is used as a decoding table, then a received vector $\underline{v}$ will be decoded correctly into the transmitted vector $\underline{u}$ if and only if the error pattern $\underline{v} - \underline{u}$ or $\underline{v} + \underline{u}$ is a coset leader[13].

For any received vector $\underline{v}$, the r component vector $r = n-k$ called the syndrome is defined as

$$\underline{s} = \underline{v}\,\underline{H}^T$$

where, $G = [I_k \quad p]$

p is k x (n - k) matrix and

I_k is k x k identity matrix

$H = [p^T \quad I_{n-k}]$ is called parity check matrix, and

$$H^T = \begin{bmatrix} p \\ I_{n-k} \end{bmatrix}$$

If $\underline{s}$ is zero, then $\underline{e} = 0$. $\underline{s}$ is different for different error patterns. Within a coset, all members have the same syndrome.

When a vector is received, its syndrome is calculated and the coset leader corresponding to this syndrome is added to the received vector to obtain the transmitted code vector $\underline{u}$ and its first b digits constitute the message vector $\underline{m}$.

In the following chapter, performance evaluation of the coding schemes considered in this chapter is carried out by simulating the channel on DEC-1090 sytem.

Chapter 3

PERFORMANCE EVALUATION OF THE CODING SCHEMES

In this chapter, the performance of the codes that are being considered is evaluated by simulating the fading channel. The model of the channel[3] that has been used for this purpose is given in the appendix. The simulation study details are given in the following section. The results obtained are illustrated through tables and graphs.

3.1 Evaluation of the Performance of the Coding Schemes

In this section, the procedure followed to evaluate the performance of error correcting codes is given. The codes that have considered are, BCH codes of length 31 and 15, single error correcting and double adjacent error-correcting codes (7,3) and (9,4) and extended Hamming codes (8,4), (8,3) and (8,2). The performance of each code is evaluated in four steps:

1. Encoding
2. Transmitting the binary sequence over the fading channel with
   (a) no explicit diversity, employing
      (i) only coding
      (ii) coding-interleaving
   (b) dual diversity, employing
      (i) only coding
      (ii) coding-interleaving

3. Demodulation

4. Decoding

3.1.1 Encoding

All these codes are systematic codes. Each error-correcting code has a unique generator matrix. In case of cyclic codes, they are also represented by generator polynomial. The random data is generated as following. A uniform random number is generated. If the generated number is greater than 0.5, it is taken as 1, otherwise 0. This k bits of 1's and 0's constitute a message vector of an (n,k) code. The message vector so obtained is multiplied by the generator matrix of dimension (k x n), which gives the code vector. Before this code vector is transmitted, zeros are converted into -1's, as BPSK modulation is considered.

At the receiver the coherent demodulation is assumed. That means the carrier amplitude and phase are known and demodulation is done using this information.

3.1.2 Transmitting the Binary Sequence over the Fading Channel

The modes of transmission used are no-explicit diversity and dual diversity. In each case, first only coding is considered and then coding along with interleaving is considered.

For no-explicit diversity case, when only coding is used, the received signal corresponding to i^{th} codeword and j^{th} bit is given as (considering coherent demodulation as mentioned earlier).

$$r_j = g_i \sqrt{E_s}\, x_{ij} + w_j \quad , j = 1,2, \ldots , n$$

where,

$$x_{ij} = +1 \text{ or } -1$$

g_i is a Rayleigh random number, which gives the channel gain. This remains the same for all bits in a codeword.

w_j is the additive white Gaussian noise, with zero mean and $N_o/2$ variance.

The procedure to compute these values of g_i and w_j are given in the Appendix I

When interleaving is used along with coding, each bit is assumed to fade independently and the signal corresponding to j^{th} bit of i^{th} codeword is

$$r_{ij} = g_{ij} \sqrt{E_s}\, x_{ij} + w_j$$
$$x_{ij} = +1 \text{ or } -1$$

g_{ij} is the channel gain, which is a Rayleigh random number and w_j is the additive white Gaussian noise with zero mean and $N_o/2$ variance.

In dual diversity mode of transmission, two signals are received for each bit transmitted. Both the received signals are independent of each other.

When only coding is used, the received signals corresponding to i^{th} codeword and j^{th} bit is represented as

$$r_{j1} = g_{i1} \sqrt{E_s}\, x_{ij} + w_j \qquad j = 1,2, \ldots , n$$
$$r_{j2} = g_{i2} \sqrt{E_s}\, x_{ij} + w_j$$

1 and 2 corresponds to the two signals received. The channel gain g_i and additive white Gaussian noise are computed similar to that of the no-explicit diversity case.

When interleaving is used along with coding the two received signals are independent and each bit fades independently

$$r_{ij1} = g_{ij1} \sqrt{E_s}\, x_{ij} + w_j$$
$$r_{ij2} = g_{ij2} \sqrt{E_s}\, x_{ij} + w_j$$

g_{ij} and w_j are computed in the same way as in the case of no-explicit diversity.

3.1.3 Demodulation

(a) No explicit diversity:

In this case, the received signal r_j corresponding to j^{th} bit of i^{th} codeword is compared against the threshold value, which is equal to zero for BPSK modulation.

$$r_j \underset{-1}{\overset{1}{\gtrless}} 0$$

If r_j is greater than or equal to 0, it is decided as 1 otherwise -1. (This can be obtained assuming coherent demodulation.)

(b) dual diversity

In this case, the maximal ratio combining is used. The two received signals of each bit are added in the ratio of their channel gain and then compared against the threshold value.

$$r_j = g_{ij1}\, r_{j1} + g_{ij2}\, r_{j2}$$

If the received value is greater than or equal to 0, it is decided as 1 otherwise -1. This method has 3 dB advantage over the no-explicit diversity method.

3.1.4 Decoding

Before a received code vector is decoded, the signals decided as -1 are taken as '0' and the rest as '1'.

BCH codes are decoded using Berlekamp algorithm, and SEC-DAEC codes and extended Hamming codes by table-look-up method, as the codeword length is very small. The decoding procedures are described in the second chapter.

The decoded message vector is compared with the actual message vector, and the number of errors occurred in each message word are counted. The bit error probability of a code for a particular SNR is obtained as the ratio of number of decoding errors to the total number of message bits.

3.1.5 Results

The bit error probability of each code is computed for a set of SNR values, for each of the four cases explained above.

In order to obtain the bit error probability of the order 10^{-4}, 10^{5} bits must be considered. Similarly to obtain the bit error probability of the order 10^{-5}, the number of bits that has to be considered is of the order 10^{6}. To obtain reliable probability of error performance of codes which are statistically significant, the amount of data that has to be tested for each value of SNR, what so ever be the case is very large.

To compute the bit error probability of the system using BCH (31,6) code for 12000 message bits, the computation time required is about 8 minutes for a SNR value of 1 dB, for each of the cases explained. However, this computation time decreases with the increase of SNR and increase of the code rate. Because of this reason, more number of message bits are tested at higher SNR values. Similarly for extended Hamming (8,4) code, to evaluate the bit error probability for 20,000 message bits, the computation time required is about 4 minutes, for a SNR value of 1 dB for each case. The computation time required for BCH codes is larger compared to that of the extended Hamming codes, as BCH decoding requires more number of computations compared to that of the Hamming codes.

The bit error probability values obtained are illustrated through tables and graphs. For low bit error rates, the results obtained are reliable, and also the curves obtained agree with the theoretical curves for low bit error values.

Table 3.1 Probability of error performance with coding and no explicit diversity (without interleaving)

(dB)	Probability for the code							
	(31,6)	(31,11)	(31,16)	(31,21)	(31,26)	(15,5)	(15,7)	(15,11)
-5	0.445	0.4317	0.4203	0.435	0.463	0.4391	0.4132	0.445
1	0.387	0.3981	0.3782	0.3809	0.391	0.3952	0.3792	0.401
3	0.357	0.338	0.3109	0.3205	0.361	0.3648	0.346	0.3721
5	0.312	0.2818	0.2685	0.270	0.323	0.2854	0.267	0.2945
7	0.2601	0.2239	0.1942	0.2091	0.274	0.2211	0.201	0.2314
9	0.2048	0.1584	0.131	0.1451	0.216	0.1542	0.143	0.167
11	0.154	0.1122	0.109	0.1192	0.163	0.1027	0.125	0.1107
13	0.1098	0.0646	0.0707	0.0865	0.112	0.0596	0.0391	0.0673
15	0.0656	0.0368	0.0501	0.0601	0.0741	0.0274	0.0236	0.0283

Table 3.2 Probability of error performance for BPSK modulation with only coding and no explicit diversity over a slowly fading channel without interleaving

$\overline{\gamma}_b$ (dB)	(8,4)	(8,3)	(8,2)	(9,4)	(7,3)	(7,4)
-5	0.4428	0.4503	0.4653	0.4416	0.4734	0.4211
1	0.3686	0.3752	0.3873	0.3525	0.396	0.329
3	0.3257	0.348	0.356	0.3172	0.367	0.278
5	0.2735	0.2819	0.295	0.2664	0.306	0.2285
7	0.2107	0.2281	0.233	0.2079	0.245	0.1723
9	0.1549	0.1726	0.1769	0.1491	0.1808	0.1232
11	0.10075	0.1216	0.1279	0.0971	0.1312	0.0791
13	0.0588	0.0763	0.0826	0.051	0.0836	0.03625
15	0.02246	0.0348	0.0455	0.0219	0.0524	0.01135

Table 3.3 Probability of error performance for BPSK modulation with coding and no explicit diversity over a slowly fading channel with interleaving

$\bar{\gamma}_b$ (dB)	(31,6)	(31,11)	(31,16)	(31,21)	(31,26)	(15,5)	(15,7)	(15,11)
-5	0.3871	0.3668	0.3402	0.3551	0.3896	0.3766	0.368	0.3827
1	0.2673	0.2422	0.2262	0.243	0.2732	0.2592	0.2484	0.2614
3	0.1991	0.1749	0.1586	0.1719	0.2011	0.1866	0.1698	0.1916
5	0.1301	0.115	0.0921	0.1193	0.142	0.1282	0.1012	0.129
7	0.072	0.05181	0.03131	0.0593	0.071	0.0606	0.0464	0.068
9	0.0436	0.02636	0.014	0.0378	0.0457	0.0324	0.0214	0.0354
11	0.0211	9.8×10^{-3}	5.01×10^{-3}	0.0199	0.0274	0.0125	8.912×10^{-3}	0.0179
13	9.66×10^{-3}	2.23×10^{-3}	1.587×10^{-3}	7.94×10^{-3}	0.0112	4.46×10^{-3}	3.019×10^{-3}	6.63×10^{-3}
15	3.16×10^{-3}	5.62×10^{-4}	3.16×10^{-4}	2.81×10^{-3}	7.46×10^{-3}	1.38×10^{-3}	9.54×10^{-4}	2.43×10^{-3}

Table 3.4 Probability of error performance for BPSK modulation with coding and no explicit diversity over a slowly fading channel with interleaving

$\bar{\gamma}_b$ (dB)	(8,4)	(8,3)	(8,2)	(9,4)	(7,3)	(8,4)
-5	0.3646	0.3784	0.3892	0.3606	0.3687	0.3452
1	0.2064	0.2232	0.2308	0.2072	0.2136	0.1913
3	0.1437	0.1627	0.1702	0.1361	C.141	0.1251
5	0.0888	0.1087	0.1120	0.09161	0.0956	0.0893
7	0.0503	0.0696	0.0703	0.0495	C.0628	0.0429
9	0.0236	0.0333	0.0436	0.0298	C.0372	0.02525
11	0.0110	0.01606	0.0177	0.C12	C.016	9.7×10^{-3}
13	4.35×10^{-3}	7.46×10^{-3}	8.19×10^{-3}	4.5×10^{-3}	58.26×10^{-3}	4.2×10^{-3}
15	1.9×10^{-3}	3.2×10^{-3}	3.7×10^{-3}	1.2×10^{-3}	3.0×10^{-3}	1.01×10^{-3}

Table 3.5 Probability of error performance for BPSK modulation with coding and dual diversity over a slowly fading channel without interleaving

$\bar{\gamma}_b$ (dB)	(31,6)	(31,11)	(31,16)	(31,21)	(31,26)	(15,5)	(15,7)	(15,
-5	0.4168	0.4067	0.3913	0.4107	0.421	0.4072	0.3874	0.412
1	0.316	0.3052	0.2898	0.2818	0.327	0.312	0.2751	0.323
3	0.2511	0.2414	0.2343	0.223	0.254	0.251	0.2281	0.263
5	0.1778	0.1672	0.1657	0.1584	0.176	0.1763	0.1557	0.178
7	0.1258	0.1155	0.0993	0.1122	0.122	0.1251	0.0892	0.128
9	0.0794	0.0622	0.0491	0.0707	0.0716	0.0792	0.0501	0.078
11	0.0446	0.0398	0.0288	0.0446	0.0423	0.0446	0.0316	0.041
13	0.0251	0.0198	0.0125	0.0251	0.0237	0.0251	0.0177	0.022
15	0.0125	8.93×10^{-3}	4.46×10^{-3}	0.0158	0.0138	0.0112	7.94×10^{-3}	0.011

Table 3.6 Probability of error performance on BPSK modulation with coding and dual diversity over a slowly fading channel without interleaving

$\bar{\gamma}_b$ (dB)	(8,4)	(8,3)	(8,2)	(9,4)	(7,3)	(7,4)
-5	C.4134	0.4201	C.4885	C.4231	0.4261	0.4038
1	C.2977	C.3039	C.3132	C.2826	C.3135	0.2791
3	C.2319	0.247	0.2593	0.2267	C.2471	C.2128
5	0.1631	0.1826	C.1964	C.1572	C.1824	C.1479
7	0.C912	C.1196	0.1275	C.0893	C.1255	C.C784
9	C.04485	0.C642	0.C752	0.C385	C.C775	C.C314
11	C.02585	0.03986	C.C485	C.C354	C.03806	0.0281
13	C.0125	0.C182	C.C253	C.C199	C.C1996	0.C134
15	4.46×10^{-3}	8.91×10^{-3}	0.0125	$8.91 \times 1C^{-3}$	$8.9 \times 1C^{-3}$	$7.C79 \times 1C^{-3}$

Table 3.7 Probability of error performance for BPSK modulation with coding and dual diversity over a slowly fading channel with interleaving

$\bar{\gamma}_b$	(31,6)	(31,11)	(31,16)	(31,21)	(31,26)	(15,5)	(15,7)	(15,11)
-5	0.381	0.3758	0.3548	0.3981	0.387	0.389	0.3528	0.379
1	0.1684	0.1572	0.1412	0.1574	0.172	0.165	0.1584	0.162
3	0.0944	0.0814	0.0794	0.0841	0.0914	0.0891	0.0891	0.0916
5	0.0446	0.0354	0.0316	0.0429	0.0473	0.0421	0.328	0.0472
7	0.0158	0.01045	7.943×10^{-3}	0.01412	0.0198	0.0138	0.011	0.0154
9	2.81×10^{-3}	1.584×10^{-3}	1.188×10^{-3}	5.046×10^{-3}	3.41×10^{-3}	3.16×10^{-3}	2.35×10^{-3}	4.32×10^{-3}
11	4.46×10^{-4}	1.995×10^{-4}	1.258×10^{-4}	1.122×10^{-3}	7.33×10^{-4}	4.25×10^{-4}	2.55×10^{-4}	7.16×10^{-4}
13	-	-	-	2.81×10^{-4}	-	-	-	-
15	-	-	-	-	-	-	-	-

Table 3.8 Probability of error performance for BPSK modulation with coding and dual diversity over a slowly fading channel with interleaving

$\bar{\gamma}_b$ (dB)	(8,4)	(8,3)	(8,2)	(9,4)	(7,3)	(7,4)
-5	0.3221	0.3308	0.3365	0.322	0.331	0.354
1	0.12535	0.1584	0.158	0.1258	0.157	0.12584
3	0.0637	0.12	0.0794	0.065	0.0794	0.0794
5	0.0197	0.0446	0.0316	0.02818	0.0398	0.0354
7	5.014×10^{-3}	0.0126	0.0112	0.010	0.0112	0.01125
9	1.125×10^{-3}	3.165×10^{-3}	1.25×10^{-3}	2.511×10^{-3}	3.162×10^{-3}	2.51×10^{-3}
11	1.99×10^{-4}	6.3×10^{-4}	4.46×10^{-4}	1.995×10^{-3}	6.309×10^{-4}	6.3×10^{-4}
13	-	-	-	6.309×10^{-3}	-	-
15	-	-	-	-	-	-

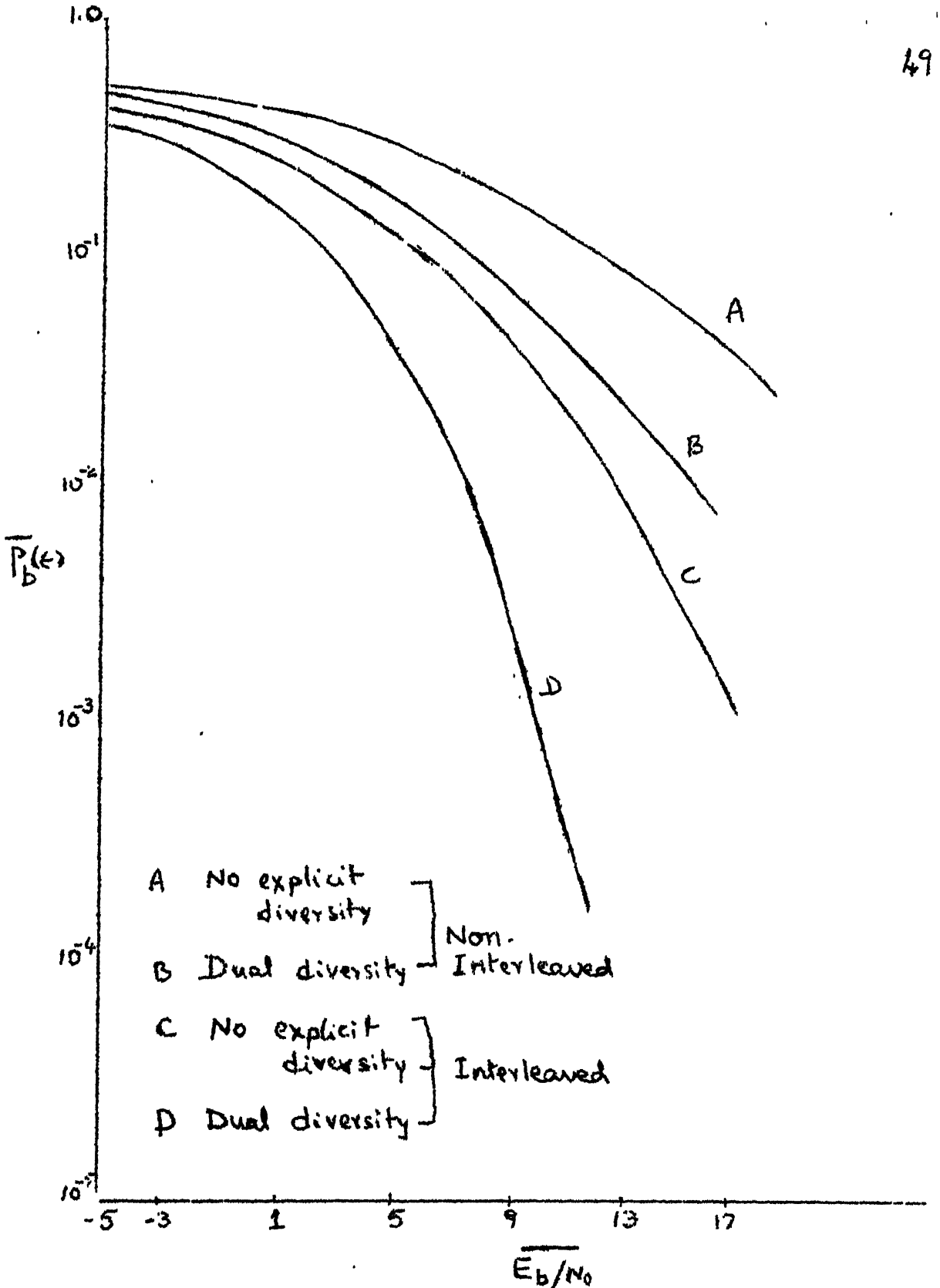


Fig 3.1. Average bit error probability for BPSK with BCH (31,6) coding over slowly fading channel

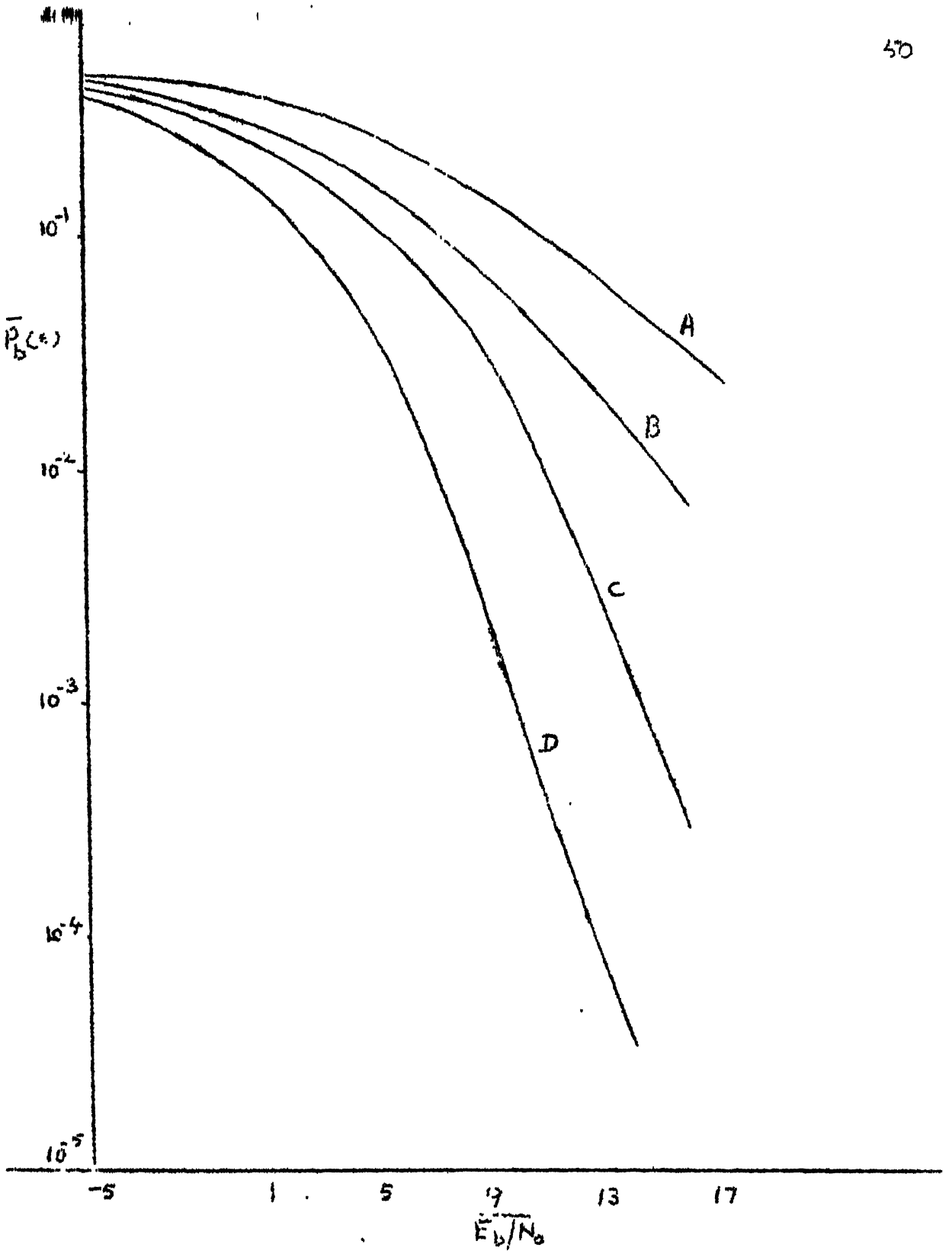


Fig 3.2. Average bit error probability for BPSK with BCH (31,11) coding over slowly fading channel

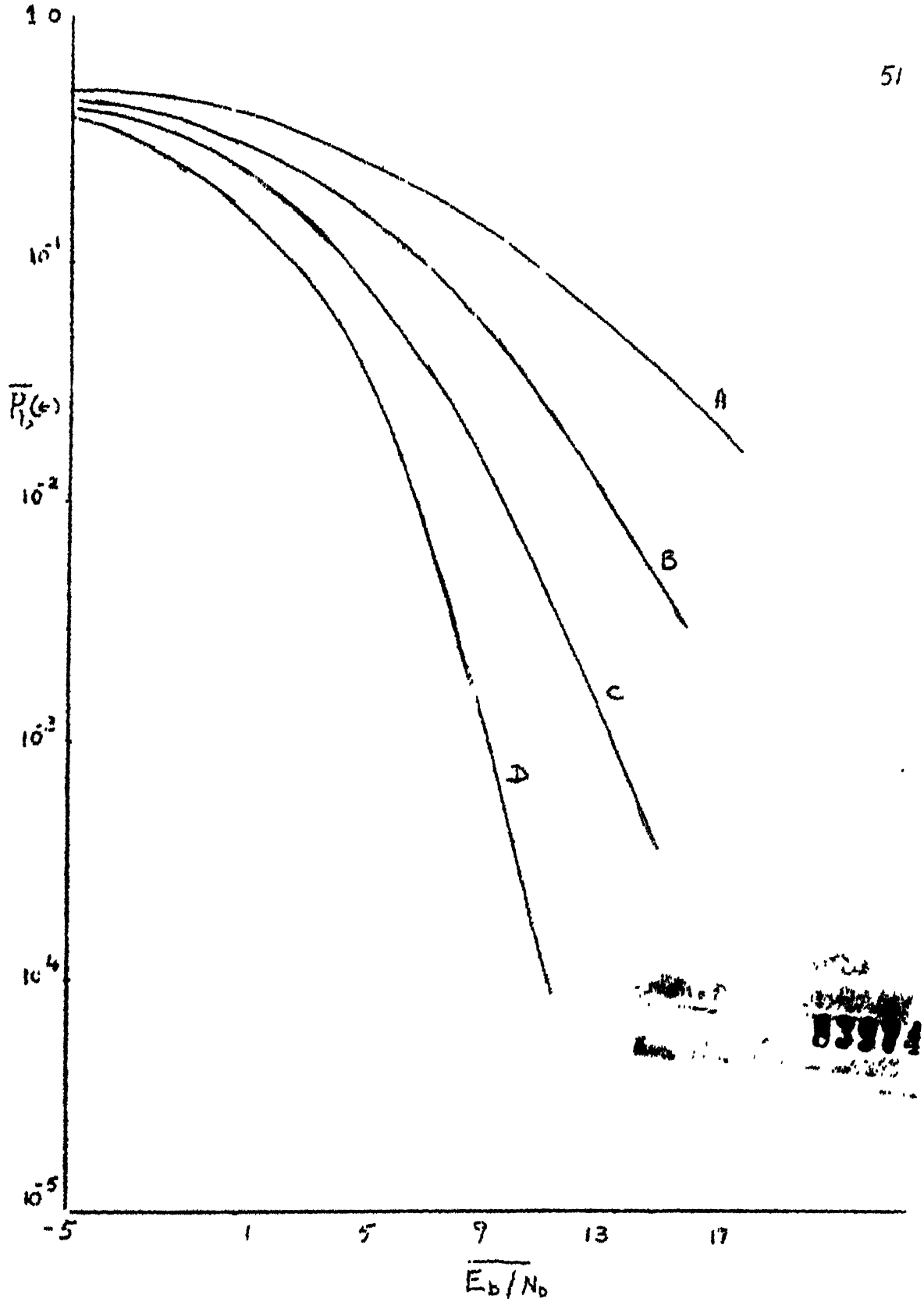


Fig 3.3 Average bit error probability for BPSK with BCH (31,16) coding over slowly fading channel

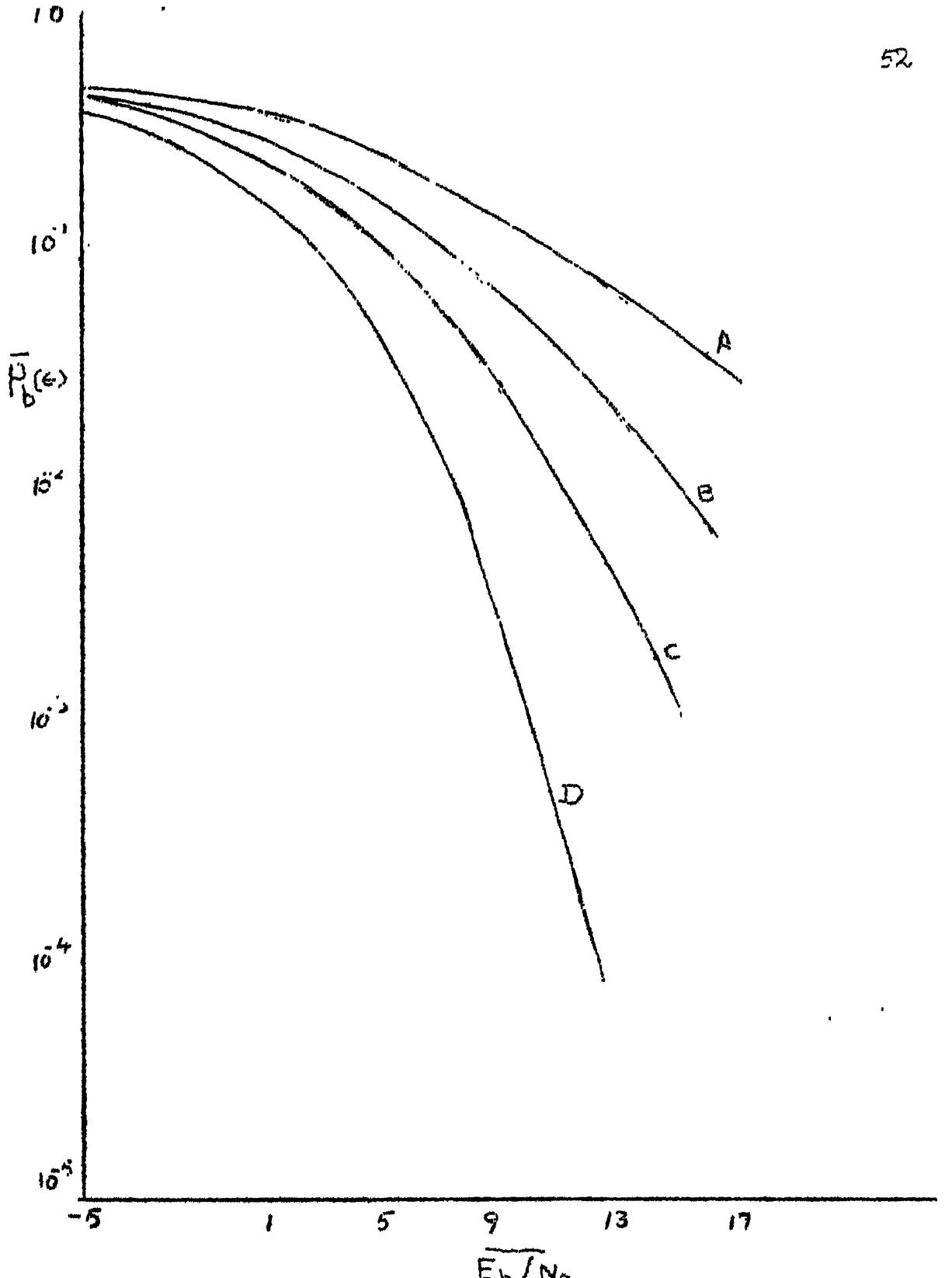


Fig. 3.4 Average bit error probability for BPSK with BCH(15,5) coding over slowly fading channel

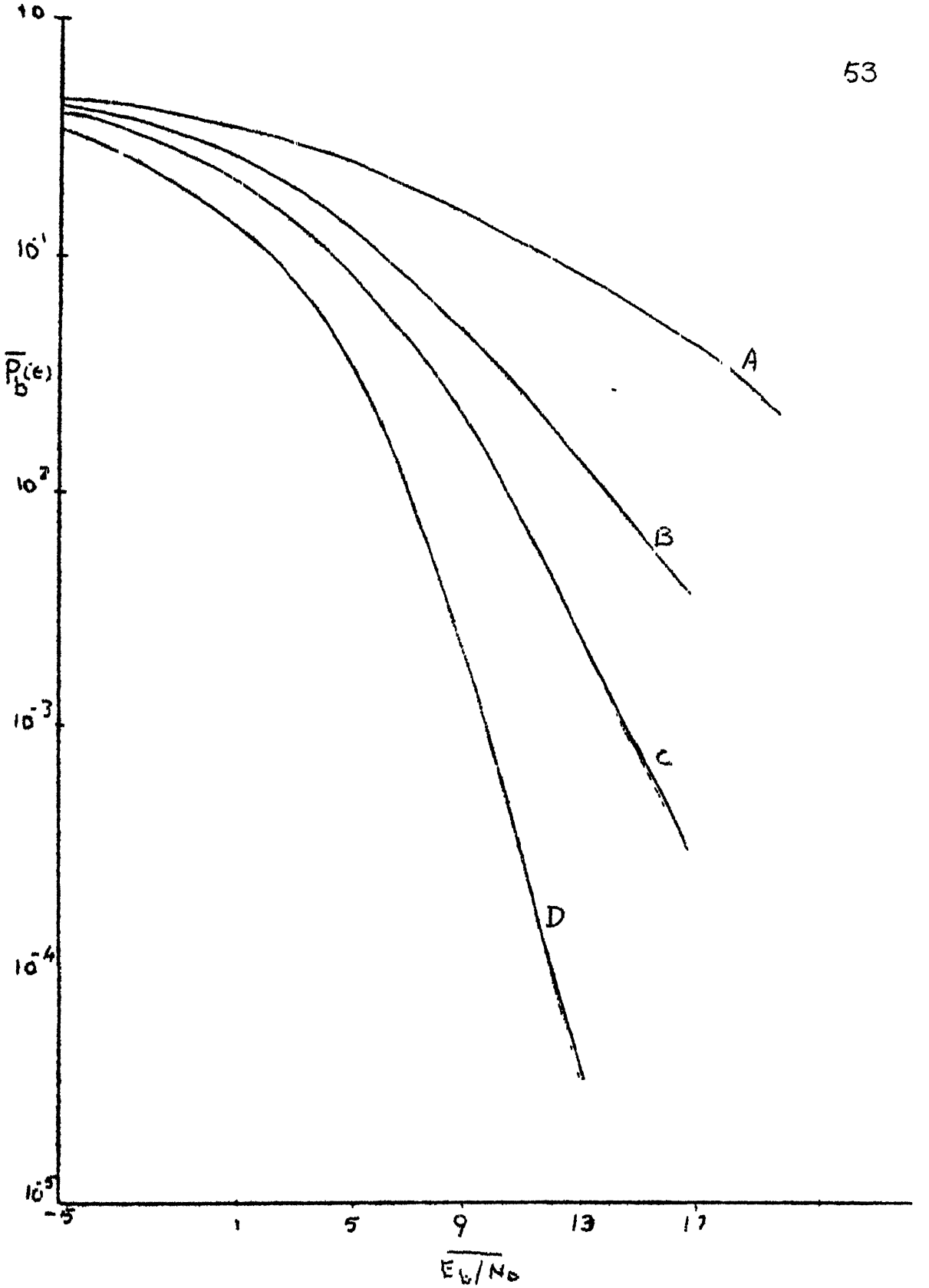


Fig 3.5 Average bit error probability for BPSK with BCH (15,7) coding over slowly fading channel

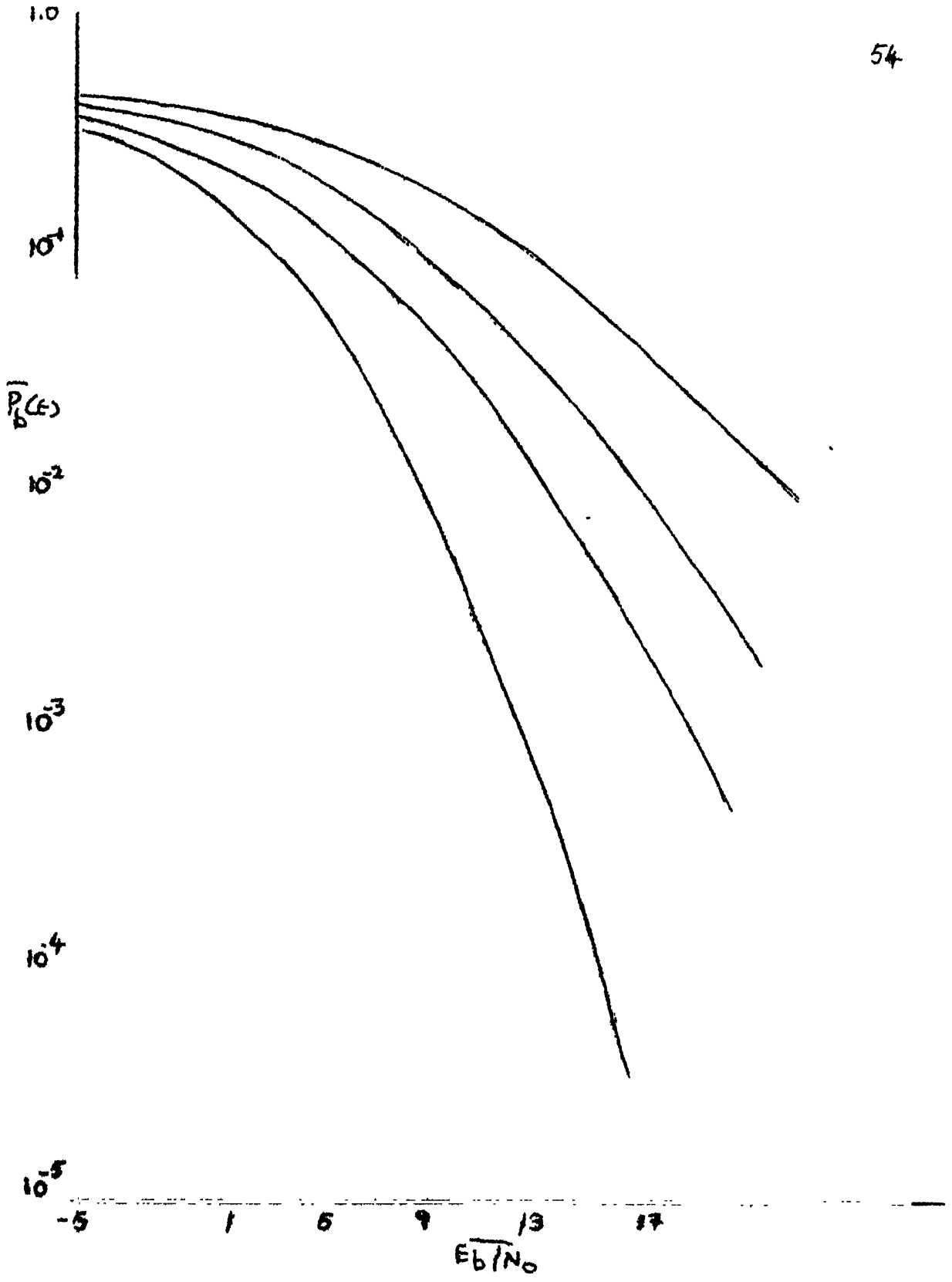


Fig 3.6 Average bit error probability for BPSK with (9,4) coding over slowly fading channel

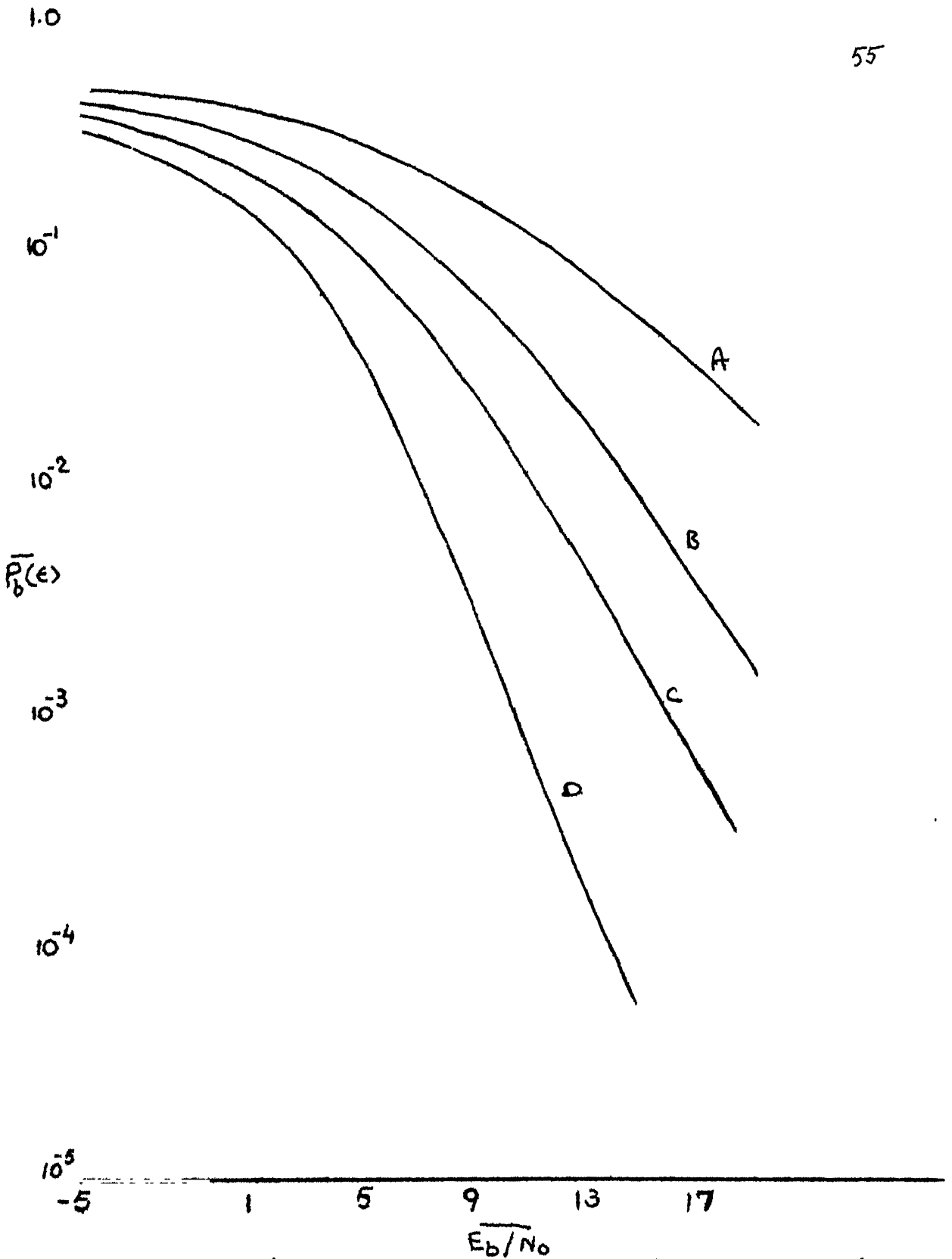


Fig 3.7 Average bit error probability for BPSK with (7,3) coding over slowly fading channel

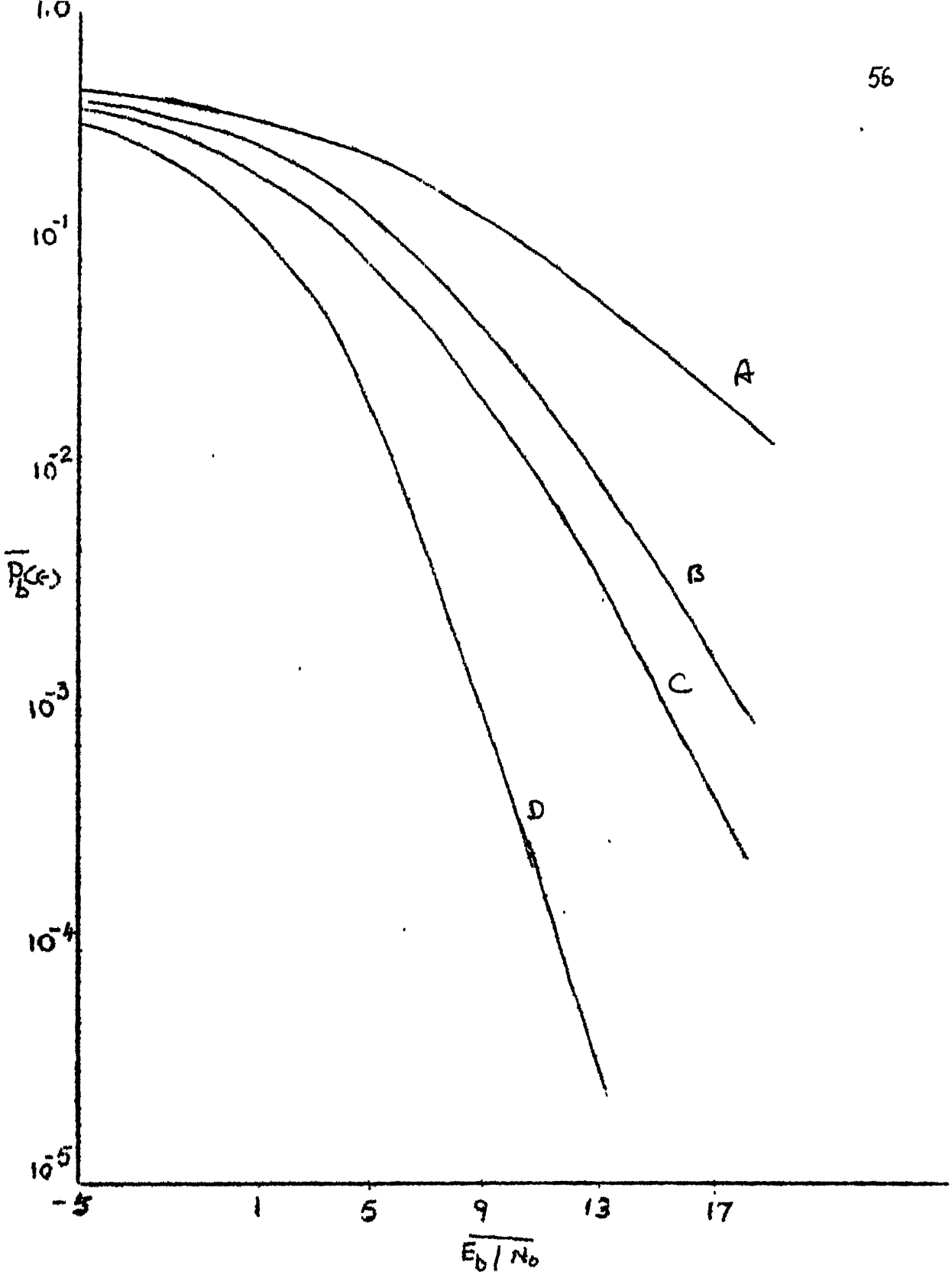


Fig 3.8 Average bit error probability for BPSK with (8,4) coding over slowly fading channel

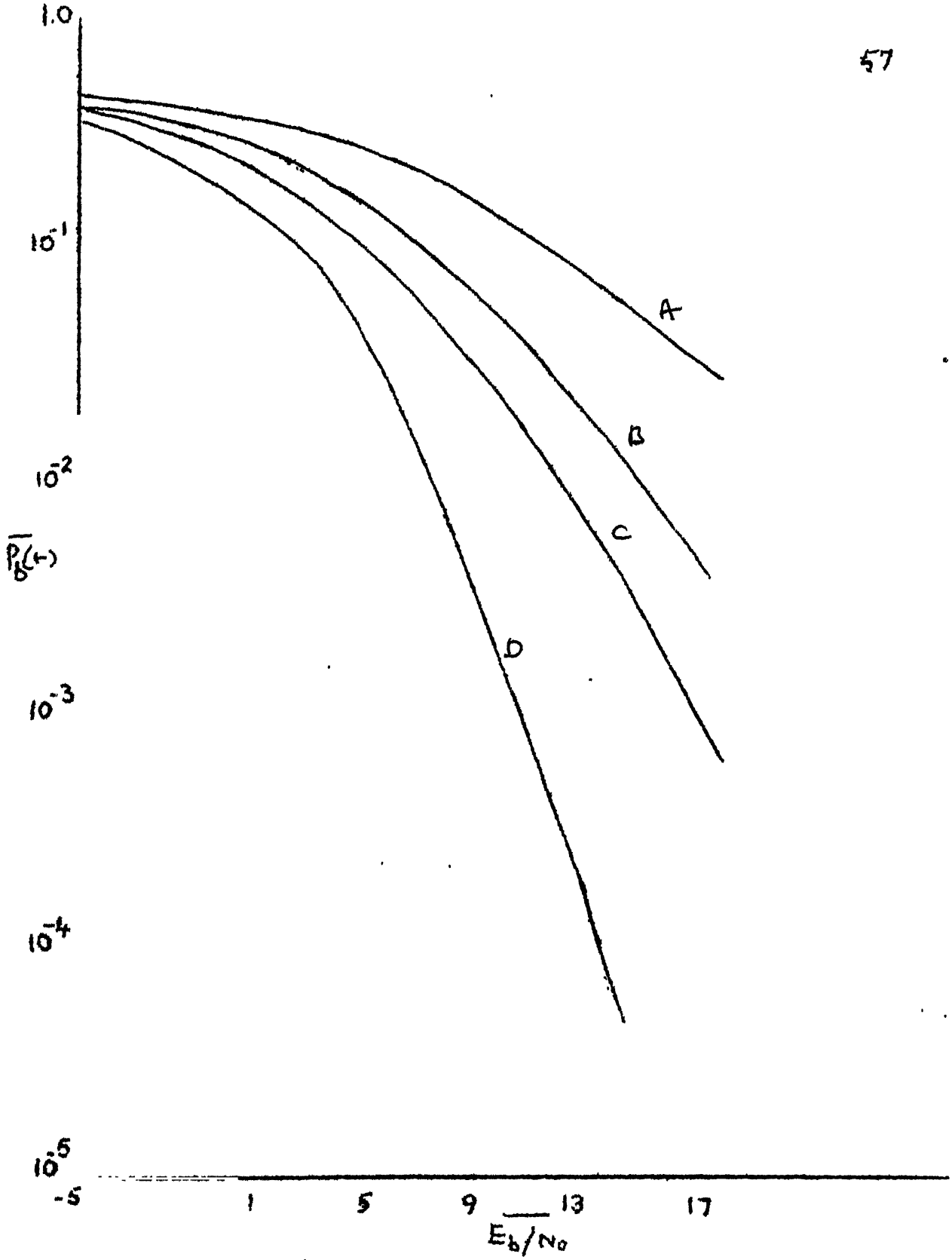


Fig 3.9 Average bit error probability for BPSK with (8,3) coding over slowly fading channel

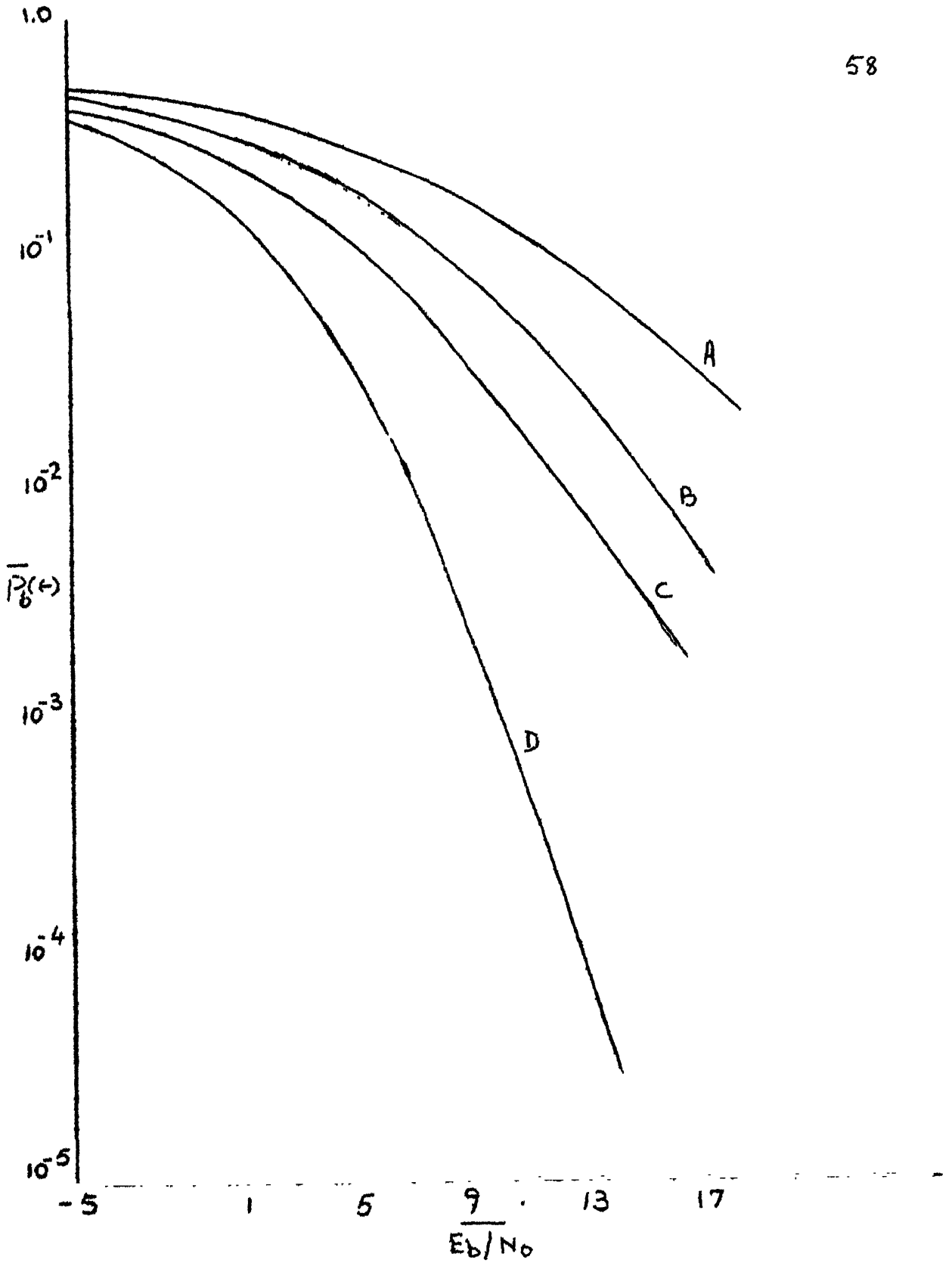


Fig 3.10 Average bit error probability for BPSK with (8,2) coding over slowly fading channel

Chapter 4

Hardware Implementation of a Codec

In Chapter 3, the performance evaluation of error-correcting coding schemes over slowly fading channel has been done through computer simulation of the channel. The advantages due to interleaving and dual diversity are also taken into consideration. The results obtained from this simulation study shows that (31,16), (31,11), (15,7), (8,4), (9,4), (8,2) and (9,3) codes achieved the required value of bit error rate for the given system specifications using interleaving and employing dual diversity. The hardware realization of BCH codes is complicated and the interleaver/deinterleaver combination is difficult. Therefore codes with smaller length are chosen for hardware implementation.

Memory requirements of interleaver/deinterleaver combination for codes under consideration are shown in Table 4.1. The amount of memory required by codes with n = 8 is 350 kbits and for n = 9, it requires an additional 100 kbits of memory. The interleaver/deinterleaver combination for code length of 8 can be implemented without much difficulty. Keeping the data rate of the channel constant, the same interleaver/deinterleaver can be used for lower rate codes (8,3) and (8,2).

Since (8,4) and (8,2) codes are of rate 1/2 and 1/4 respectively, it is easy to implement these schemes.

Table 4.1

Memory Required for Interleaver/Deinterleaver

Code	Memory Required (kbits)
(7,4)	525
(7,3)	350
(8,4)	350
(8,4)	450
(15,5)	700
(15,7)	720
(15,11)	954.5
(31,6)	1107
(31,11)	845.45
(31,15)	969.75
(31,21)	1107
(31,25)	1278

Consequently, it has been decided to implement (8,4) and (8,2) coding schemes. (8,4) is used for high data rates and (8,2) is for low data rates.

In this chapter hardware implementation details of (8,4) and (8,2) codes are presented. Description of the encoder and the decoder, and their implementation are given in the first and the second sections respectively. The block diagram of the digital communication system employing a codec is shown in Fig. 2.1.

4.1 Description of the Encoder

Message words of k bits are encoded into an (n,k) code by generating (n-k) parity check bits from the parity check equations. These parity check bits are transmitted along with the message bits.

The parity check equations of (8,4) code, assuming $\underline{x} = [x_1, x_2, x_3, x_4]$ to be the message vector, are

$$z_1 = x_1 + x_2 + x_4$$
$$z_2 = x_2 + x_3 + x_4$$
$$z_3 = x_1 + x_2 + x_3$$
$$z_4 = x_1 + x_3 + x_4$$

The codeword $\underline{y} = [x_1, x_2, x_3, x_4, z_1, z_2, z_3, z_4]$

For (8,2) code, with $\underline{x} = [x_1, x_2]$, the parity check equations are

$$z_1 = 0;\ z_2 = x_1;\ z_3 = x_2;\ z_4 = x_1 + x_2;\ z_5 = x_1;\ z_6 = x_2$$

The codeword $\underline{y} = [x_1, x_2, z_1, z_2, z_3, z_4, z_5, z_6]$

(8,4) code is employed for high data rates (i.e., 1 Mbps) and (8,2) code is employed for low data rates (i.e., 500 kbps).

The channel data rate is 2048 kbps for both cases. Codewords corresponding to (8,4) and (8,2) codes are given along with the message words in Table 4.2.

Table 4.2

(8,2) and (8,4) codewords

(8,2)		(8,4)	
Message Word	Code Word	Message Word	Code Word
00	00000000	0000	00000000
01	01001101	0001	00011101
10	10010110	0010	00100111
11	11011011	0011	00111010
		0100	01001110
		0101	01010011
		0110	01101001
		0111	01110100
		1000	10001011
		1001	10010110
		1010	10101100
		1011	10110001
		1100	11000101
		1101	11011000
		1110	11100010
		1111	11111111

4.1.1 Implementation of the Encoder

The block schematic of the encoder is given in Fig. 4.1. The encoder consists of a serial to parallel converter, latches, encoding circuit and a parallel to serial converter.

The serial binary data to be encoded is converted into words of 4 bits (for (8,4) code) or 2 bits (for (8,2) code) depending on the code employed using the serial to parallel converter. Clock frequency of this serial to parallel converter is (k/n) x 2 MHz.

Whenever the TLT signal goes high a message word is latched into the encoding circuit (consisting of EX-OR gates) which

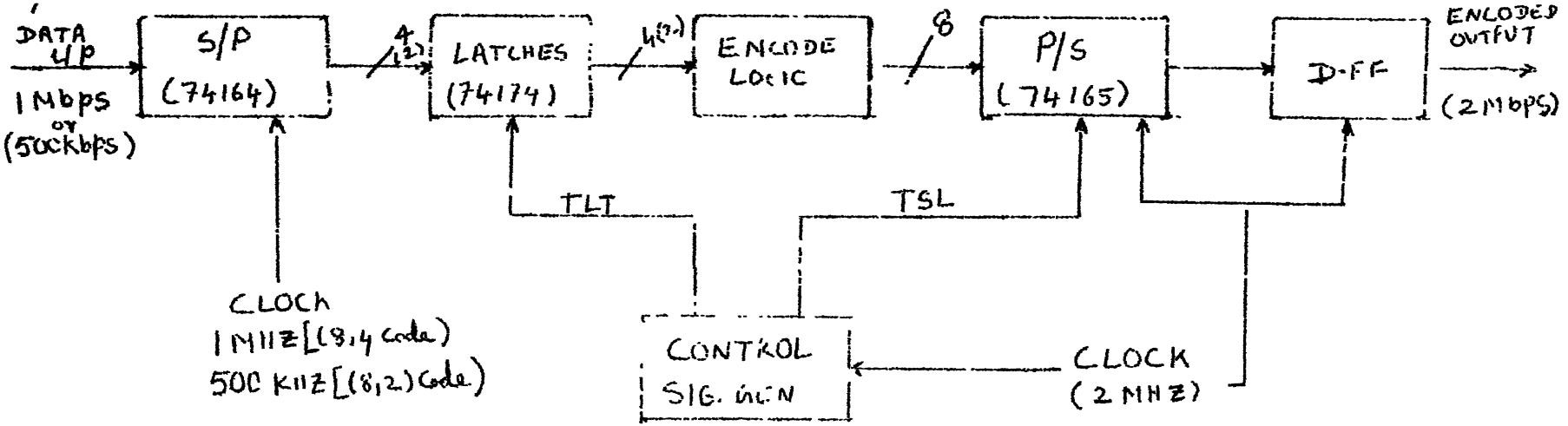


Fig 4.1. ENCODER BLOCK DIAGRAM.

generates parity check digits. Message bits are delayed by an equal amount of time as taken by the EX-OR gates so that all bits of a codeword appear at the output at the same time.

This encoding circuit is followed by a parallel to serial (P/S) converter. Encoded word is loaded into P/S converter whenever TCL signal goes low. P/S converts a word of 8 bits into a sequence of 8 bits. The clock frequency of P/S converter is 2 MHz.

The output of the P/S converter is given to a D-FF in order to align the data with the clock properly. This D-FF outputs the encoded data serially at a rate of 2 Mbps.

4.1.2 Clock Generator

A simple RC oscillator circuit is used to generate the clock at a frequency of 2 MHz. Circuit diagram of clock generator is given in Fig. 4.2.

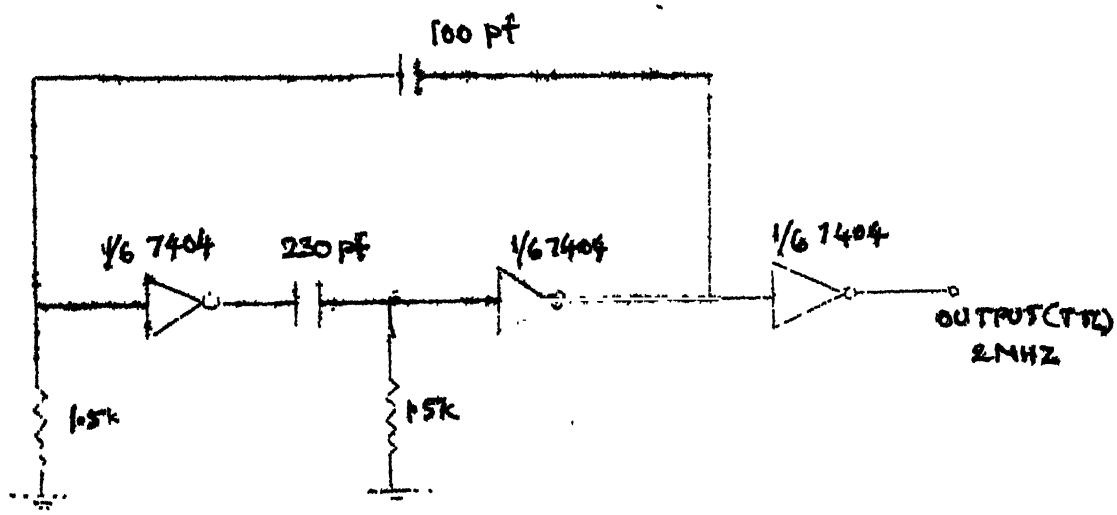


Fig. 4.2

All control signals are derived from this clock. Timing diagram of clock and control signals is given in Fig. 4.3.

TLT - Transmitter latch signal (for the latches after S/P)

TSL - Transmitter shift-load signal (for the P/S)

4.2 Description of the Decoder

Data received from the deinterleaver at a rate of 2 Mbps is given to the decoder. Decoding of the received words is done by table-look-up method. The decoding table is stored in an EPROM (2716) and this reduces the hardware complexity and increases the reliability. The maximum data rate that can be attained using EPROM is 24 Mbps. The decoder circuit consists of a S/P converter, latches, EPROM and a P/S converter. The block schematic of the decoder is given in Fig. 4.4.

4.2.1 Implementation of the Decoder

The received data sequence is converted into words of 8 bits using a S/P converter. Whenever the TLT signal goes high, a received word is latched onto the address inputs of EPROM which in turn gives out the decoded message bits corresponding to the received word. The decoded output bits are taken from O_3 - O_6 of EPROM (for (8,4) code), and O_1 - O_5 of EPROM (for (8,2) code). Whenever the TSL signal goes low, the outputs are loaded into the P/S converter. This output of P/S converter is given to a D-FF in order to align the decoded data properly with the clock. The decoded bits are taken out serially at a rate of (k/n) x 2 Mbps.

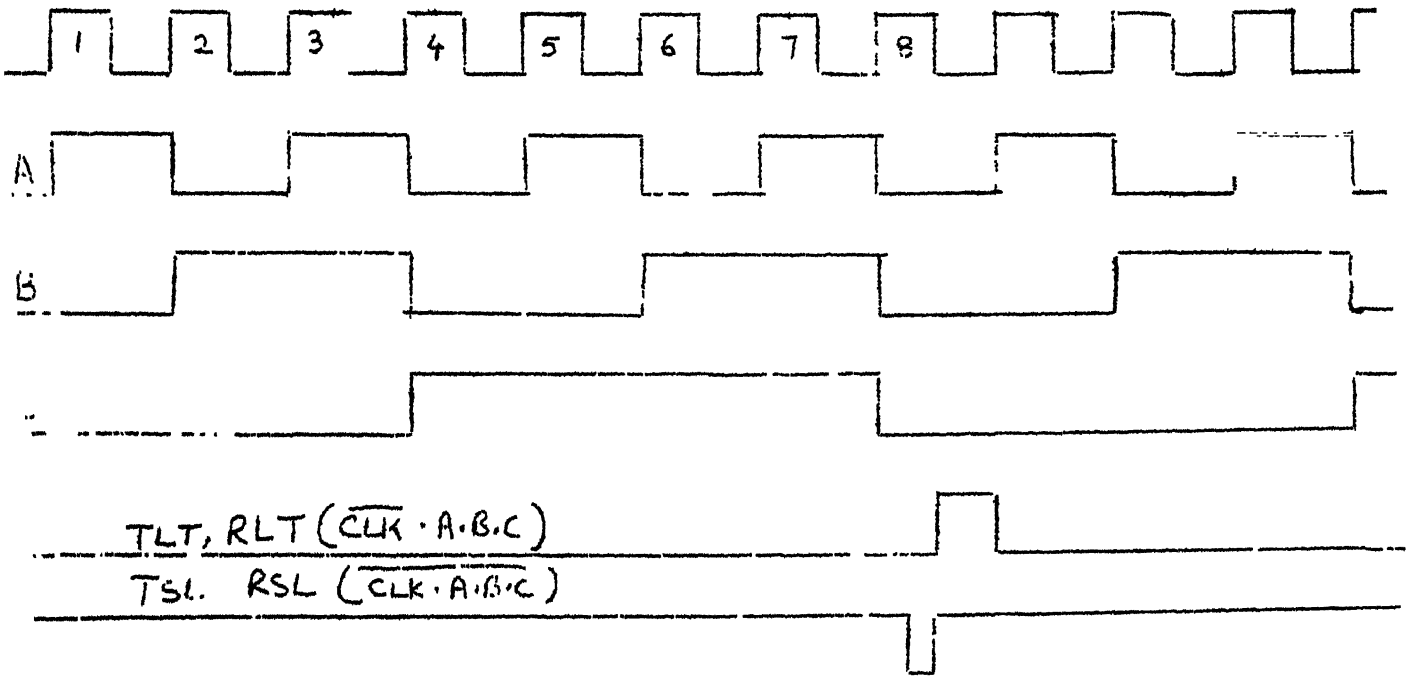


FIG. 4.3 TIMING DIAGRAM.

The RLT and RSL signals are generated separately at the decoder.

RLT - receiver latch signal (for latches after S/P)

RSL - receiver shift-load signal (for P/S)

4.2.2 Word Synchronizer

Word synchronization is necessary in order to ensure that all bits in a word of 8 bits, to be decoded correspond to the transmitted word, i.e., latching at the output of S/P must be done precisely after receiving the last bit of a word of 8 bits that is being transmitted. Word synchronization can be achieved by making use of the redundancy of the code as explained below.

Every memory location in the EPROM (2716) used, can store a data word of 8 bits. The message word that is being stored in each memory location addressed by the code vector (which may be correct or erroneous) if of maximum length 4, and the most significant 4 bits are not used. By making use of this redundancy, word synchronization can be achieved.

Zeros are stored in the most significant four bit locations addressed by all correct codewords and those with single errors. Ones are stored for the rest of the words. If there are 64 or erroneous words (excluding the ones with single errors) in 256 consecutive words received, then it is understood that the decoder is out of synchronization. Word synchronization is done

by clock slipping and thereby delaying the latching instant by one clock at the output of the S/P.

As shown in the Fig. 4.4, the most significant output bit of the EPROM is continuesly monitored by the word synchronization circuit. Whenever this bit goes high, error counter count is incremented. If there are 64 or more erroneous words (excluding those with single errors) in 256 consecutive words, clock slipping circuit is enabled after resetting the error count to zero.

This clock slipping circuit generates a signal which is normally at logical '1' and goes low for one clock period whenever clock slipping circuit is enabled. This signal is ANDed with the clock and the resultant signal is used to generate PLT and PSL signals. Now latching signal misses one clock cycle and thus moves back by one clock period. This continues until the word sync. is established. The decoder should be in word synchronization within 7 trials (worst case).

This circuit not only achieves word synchronization but monitors the sync. Whenever the decoder goes out of synchronization it is automatically brought back.

The complete encoder-decorder circuit is tested using PRBS sequence.

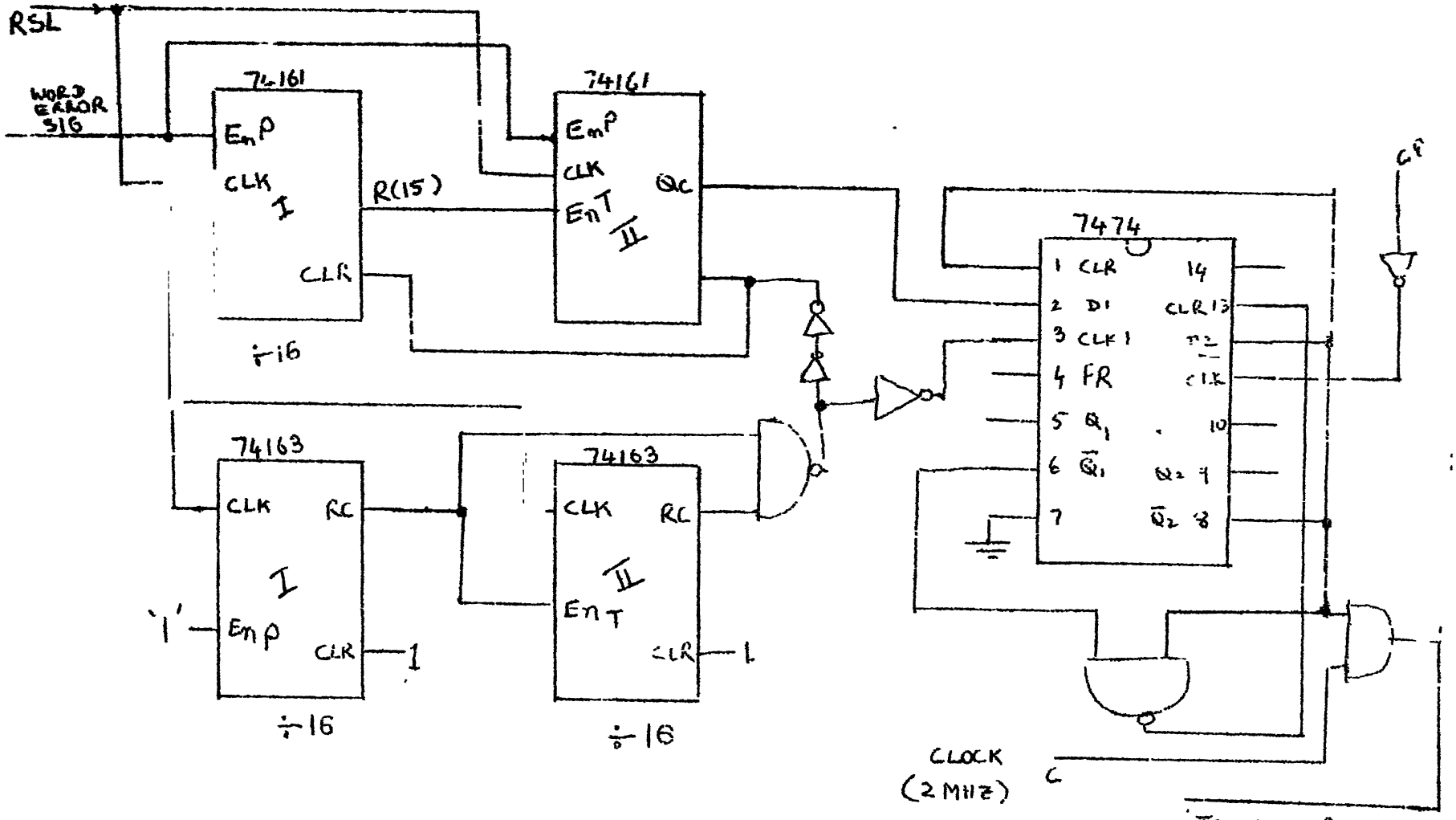


FIG. 4.5. SCHEMATIC OF WORD SYNCHRONIZER.

CONCLUSIONS

In this thesis, an attempt has been made to find some error correcting coding schemes, which decrease the value of (E_b/N_o) required to achieve the specified average bit error probability for a typical mobile trops system specifications. The error correcting coding schemes that have been considered are BCH codes over GF (2^5) and GF (2^4), single error correcting and adjacent double error correcting codes (modified Hamming codes) (9,4) and (7,3), and extended Hamming codes (8,4), (8,3) and (8,2). The bit error probability achieved by the system using the above mentioned codes over a slowly fading channel has been evaluated for the cases of no-explicit diversity and dual diversity with and without interleaving, by means of computer simulation for a total computational time of 6 hours.

The input data rate of the system is 1 Mbps. When an (n,k) code is used, the channel data rate is (n/k) Mbps. The typical fade duration of the channel is 100 msec. Whenever a fade occurs, the member of bits caught in a fade is approximately (1/10) (n/k) Mbits. This value is significantly large for high data rates. In order to obtain reliable probability of error performance of codes which are statistically significant, the amount of data that has to be tested for different SNR values is very large. Such large amounts of data require very high values of computation time.

The main purpose of interleaving the encoded data is to allow different bits of a codeword to fade independently. If the interleaving and deinterleaving are to be implemented on a digital computer by storing the data as shown in Fig. 2.2, the amount of storage required is very large and organizing such large amounts of data is difficult. This can be obviated by taking independent channel gains for each bit of a codeword thereby realizing the effect of interleaving.

When a BCH (31,6) code is used to evaluate the bit error probability of the system for a sample data of 12000 message bits, the CPU time required for encoding these message bits, passing the coded bits over the channel and decoding the received bits is about 3 minutes for a SNR value of 1 dB, for any case of diversity with or without interleaving. This computation time decreases with the increase of SNR, as the number of channel bit error decreases. As the code rate increases, the error correcting capability decreases, the number of computations involved decreases thereby reducing the computational time. Similarly for extended Hamming (8,4) code, to perform the operations stated above for 20,000 message bits of data, the computation time required is 4 minutes for a SNR value of 1 dB, this decreases with the increase in SNR. The decoding procedure of extended Hamming codes is very simple compared to that of the BCH codes. Because of the large values of computation time, the amount of data tested is very less compared to the required value.

Within these practical limitations, based on the results obtained from this simulation study, a few conclusions can be drawn. First, coding schemes without interleaving are not worth considering. Second, the value of (E_b/N_o) required to achieve the specified bit error probability can be decreased using multi-diversity techniques. It has been observed that for low bit error rates, the results obtained agree with the theoretical results. The problem of handling large amount of data and computational time can be circumvented by using a channel simulator and the codec-interleaver combination corresponding to the code under consideration.

From the simulation study, it has been found that (31,6), (31,11), (31,16), (15,5), (15,7), (8,4), (8,4), (8,2) and (8,3) codes achieve the required value of (E_b/N_o) with interleaving and using dual diversity. But the hardware implementation of longer codes is difficult and the interleaver/deinterleaver combination is complex. Consequently, error correcting codes of smaller codelength are chosen for hardware implementation.

The extended Hamming (8,4) code is chosen for implementation because of the following reasons. First, the performance of (8,4) code is comparable to that of the (8,4) code. Second, the codec for (8,4) code is simpler compared to that of (8,4) code. Third, the (8,4) code requires 100 bits of more memory for interleaver and deinterleaver structure compared to that of the (8,4) code.

It has been observed that (8,2) code achieves the required probability of error rate. As it can be used for low data rates (500 kbps), it has been implemented. Moreover, the channel data rate of (8,2) code is same as that of (8,4) code. Therefore, the same interleaver/deinterleaver combination of (8,4) can also be used for (8,2) code.

The (8,3) extended Hamming code achieves the specified bit error rate for the given specifications. It is capable of correcting all single errors, all adjacent double and triple errors, and a few other random errors. Consequently by employing (8,4), (8,3) and (8,2) coding schemes together, and using the same interleaver and deinterleaver combination, one can achieve the specified probability of error rate for data rates ranging from 500 kbps to 1 Mbps.

REFERENCES

1. M. Schwartz, W.R. Bennet and S. Stein, 'Communication Systems and Techniques', McGraw-Hill, N.Y., 1966.

2. S.U.K. Pillai, 'Feasibility Study of Modems with Soft Decision Decoding for Digital, M.Tech. Thesis, July 1982.

3. B.V. Rao, 'An Interleaver for Digital Tropo and System Performance Evaluation', M.Tech. Thesis, Sept. 1984.

4. J.D. Rogers, 'Introduction to digital tropo for military tactical communication', Communication and Broadcasting, vol. 6, No.3, pp 3-9.

5. J.D. Rogers, M.H. Stears and D.W. Baker, 'Radio Equipment for Digital Tropo Military Tactical Communication', Communication and Broadcasting, vol. 7, No.1.

6. P.A. Bello, L. Ehrman, T.H. Crystal, 'Tropo-scatter multi-channel digital systems study', Technical Report No. RADC-TR-67-218, Signatron Inc.

7. W.W. Peterson, 'Error-Correcting Codes', M.I.T. Press, Cambridge, 1961.

8. W.W. Peterson and E.J. Weldon Jr., 'Error-Correcting Codes', M.I.T. Press, Cambridge, 1972.

9. N.M. Abramson, 'A class of systematic codes for non-independent errors', IRE Trans. on Information Theory, IT-5, No.5, pp 150-157, Dec. 1959.

10. J.E. Meggit, 'Error-correcting codes and their Implementation for data transmission systems', IRE Trans. on IF, IT-7, No.4, pp 234-245, Oct. 1961.

11. Shu Lin, 'An Introduction to Error-Correcting Codes', Prentice-Hall, Inc., NJ, 1970.

12. E.R. Berlekamp, 'Algebraic Coding Theory', McGraw-Hill, NY, 1968.

13. J.M. Wozencraft and I.M. Jacobs, 'Principles of Communication Engineering', John Wiley Inc., N.Y. 1965.

14. G. Longo, 'Algebraic Coding Theory and Applications', Springer-Verlag, 1979.

15. P. Mowen and S.H. Richman, 'Adaptive Data Transmission Study', Technical Reports, Signatron Inc.

16. J.F. Pieper, J.G. Proakis, R.R. Reed and J.K. Wolf, 'Design of efficient coding and modulation for a Rayleigh fading channel', IEEE Trans. on IT, vol. IT-24, No.4, Jul 1978, pp 457-468.

17. K. Brayer, 'Error Correction Codes', IEEE Trans. on Com Tech., vol. COM-19, Part II, pp 781-781, Oct. 1971.

18. Peterson, W.W., 'Encoding and Error-correcting Procedures for the Bose-Chaudhuri Codes', IRE Trans. on Information Theory, IT-6, pp 459-470, Sep. 1960.

19. Bartee, T.C., and D.I. Schneider, 'An Electronic Decoder for Bose-Chaudhuri - Hocquenghem Error Correcting Codes', IEEE Trans. on Information Theory, IT-8, pp 17-24, Sept. 1962.

20. Berlekamp, E.P., 'On Decoding Binary Bose-Chaudhuri - Hocqueughem Codes', IEEE Trans. on Information Theory, IT-11, pp 577-580, Oct. 1965.

21. Forney, G.D., 'On Decoding BCH Codes', IEEE Trans. on Information Theory, IT-11, pp 549-557, Oct. 1965.

Appendix

Description of the slowly fading channel

The dessription of the channel that has been simulated is given below. A non-dispersive, Rayleight fading channel is simulated.

When the fade rate is very very smaller than the data rate as is the case in the problem considered, where the data rate is of the order of 1 MHz and the fade rate of about 10 Hz, the channel gain can be assumed to be constant for many bit durations.

Assuming BPSK modulation and coherent demodulation at the receiver, the received sample

$$r = \begin{cases} g\sqrt{E_s} + \omega & : H_1 \\ -g\sqrt{E_s} + \omega & : H_0 \end{cases}$$

where $|g|$ is Rayleigh distributed random variable and ω is a zero mean white Gaussian random variable with variance $N_o/2$. $|g|$ is obtained as the magnitude of a complex Gaussian random variable g. The channel is slowly fading and the complex gain 'g' remain the same for many bits and this causes the performance to be poor whenever deep fades occur.

'g's' are the sampled values of the Gaussian random process g(t), with zero mean and autocorrelation function R( ). The ACE of sampled process 'g' is also R( ) where takes on only the values of multiples of the sampling interval.

In a troposcatter channel, where the fade rate is about 10 Hz, R( ) almost vanishes after 100 msecs. The gain g should be obtained at the rate of $_c$ seconds. Ten samples are generated in 100 msecs duration and interpolation is done to get the sampled values in between. This method is used to reduce the computation time and the error negligible.

For this the sampled ACE is R_n and R_n should be chosen such that it is negligible after n > 10. It would be realistic to assume that R_n decreases in the interval $0 \leq n \leq 10$ with increasing n and R_o is taken to be unity so that 'g' has unity power. To get these correlated random numbers in auto-regressive (AR) model is used.

$$g_n = a_1 g_{n-1} + a_2 g_{n-2} + e_n$$

where e_n is independent zero mean Gaussian random variable with variance 1.

It can be shown that g_n have an ACE given by

$$R_n = a_1 R_{n-1} + a_2 R_{n-2}$$

a_1 and a_2 are selected such that the ACF has the kind of shape as expected above. $a_1 = 0.4$ and $a_2 = 0.2$ sent the ACF required. Linear interpolation is used to get the required number of samples in between. Using this method the required correlated Rayleigh numbers are generated.

The additive white Gaussian noise samples are generated with a variance depending on the SNR required.

When the channel bits are not interleaved, the above procedure is used to generate the channel gain. The effect of interleaving is to make each bit of a codeword to fade independently. So an independent channel gain is considered for each bit of a codeword.